

北京科技大学实验报告

学院： 计算机与通信工程学院

专业： 计算机科学与技术

班级： 计 184

姓名： 王丹琳

学号： 41824179

实验日期： 2020 年 11 月 15 日

实验名称：汇编预习作业

实验目的：通过自学汇编语言编程，完成预习报告，为实验课奠定基础。

实验要求：通过 CG 系统抽取预习题目，在到实验室上课前完成预习报告。

坚决杜绝抄袭！

实验内容：

1、简答题

题目 1：

8086/8088CPU 中有几个变址寄存器？分别简要描述它们的功用是什么？这其中是否有可作为地址寄存器使用的？

答：

- 8086/8088CPU 中有 2 个变址寄存器，分别为 SI、DI
- SI 是源操作数变址寄存器，用来存放源数据的地址
DI 是目的操作数变址寄存器，用来存放目的数据的地址
- SI、DI 都可作为地址寄存器使用

2、编程题

要求：按题目中的要求完成 2 道编程练习题。

所用编程工具：

DOSBox-0.74-3

Visual Studio Code（其中包括插件 masm-code v1.2.8，MASM/TASM）

编程题 1：

用汇编语言编写程序，实现以下功能：

假设 15 名学生微机接口课程的成绩为：95、63、89、90、69、70、77、100、59、60、61、0、43、25、89。编程统计各分数段的人数（分数段划分：60 分以下，60-69, 70-79, 80-89, 90 及以上），在屏幕上显示统计结果。

请在答题时，按照以下格式填写内容：

1、程序源代码（包含必要的注释信息）

2、程序运行截图

1) 程序源代码

```
stack segment                ;堆栈段
    dw 128 dup (0)
stack ends

data segment                 ;数据段
    result0 db 0             ;<60
    result1 db 0             ;[60,70)
    result2 db 0             ;[70,80)
    result3 db 0             ;[80,90)
    result4 db 0             ;[90,100]
    score db 95,63,89,90,69,70,77,100,59,60,61,0,43,25,89
    outbuf db 200 dup(0)
data ends

code segment                 ;代码段 TODO: 根据成绩, 对
应 result++
    assume cs:code,ds:data,ss:stack

start:
    ; 初始化
    mov ax,data
    mov ds,ax
    mov sp,128

    mov si,offset score
    mov cx,15                ;cx 循环次数(一共有 15 个成绩, 循环 15 次)
loop1:
    ;开始根据成绩进行判断
    mov bx,15
    sub bx,cx
    mov al,[si+bx]           ;ax 存放当前学生成绩
    mov ah,00h
    mov dx,59                ;bx 作为比较数
```

```

        cmp    ax,dx
        ja     s1                ;ax>59 则跳转
        inc    result0
        jmp    loop1_end

s1:
        mov    dx,69             ;bx 作为比较数
        cmp    ax,dx
        ja     s2                ;ax>69 则跳转
        inc    result1
        jmp    loop1_end

s2:
        mov    dx,79             ;bx 作为比较数
        cmp    ax,dx
        ja     s3                ;ax>79 则跳转
        inc    result2
        jmp    loop1_end

s3:
        mov    dx,89             ;bx 作为比较数
        cmp    ax,dx
        ja     s4                ;ax>89 则跳转
        inc    result3
        jmp    loop1_end

s4:
        inc    result4
;判断结束
loop1_end:
        loop   loop1

        mov    bx,0
        mov    ch,0

;将 result0,result1,result2,result3,result4 输出
output:
        mov    ah,0
        mov    al,result0
        call   putInt            ;子函数调用
        mov    al,' '
        call   putchar          ;子函数调用

        mov    ah,0

```

```

        mov     al,result1
        call    putInt                ;子函数调用
        mov     al,' '
        call    putchar              ;子函数调用

        mov     ah,0
        mov     al,result2
        call    putInt                ;子函数调用
        mov     al,' '
        call    putchar              ;子函数调用

        mov     ah,0
        mov     al,result3
        call    putInt                ;子函数调用
        mov     al,' '
        call    putchar              ;子函数调用

        mov     ah,0
        mov     al,result4
        call    putInt                ;子函数调用
        mov     al,' '
        call    putchar              ;子函数调用

        mov     ah,4ch
        int     21h

putchar:                                ; 把 dx 地址中的字符输出
        push    dx
        mov     dx,ax
        mov     ah,02h
        int     21h
        pop     dx
        ret

putInt:
        push    si
        push    dx
        mov     dx,0
        mov     si,offset outbuf
        call    dtoc
        mov     dx,offset outbuf
        mov     ah,9
        int     21h
        pop     dx

```

```

        pop     si
        ret

dtoc:                                     ; 逐次取余数，并入栈，最后
再倒序拿出来，在 buffer 最后放 0
        push    ax
        push    di
        push    cx
        push    dx
        push    si
        mov     di,0                     ; 记录入栈次数
dtoc_s1:
        mov     cx,10
        mov     dx,0
        div     cx

        mov     cx,ax                     ; 如果商为 0，则求值完成
        jcxz    dtoc_s2
        add     dx,30h
        push    dx                       ; 求得的 ASCII 码入栈
        inc     di
        jmp     dtoc_s1
dtoc_s2:
        add     dx,30h
        push    dx
        inc     di
        mov     cx,di                     ; cx 为转化后的字符串长度
dtoc_s3:
        pop     ax
        mov     [si],al
        inc     si
        loop    dtoc_s3                  ; 将 ASCII 码出栈
        mov     al,"$"
        mov     [si],al                  ; 最后一位放$(虽然本来初始
化的时候就已经是 0 了)
        pop     si
        pop     dx
        pop     cx
        pop     di
        pop     ax
        ret

code ends
end start

```



3) 程序运行结果的屏幕截图

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
To activate the keymapper ctrl-F1.
For more information read the README file in the DOSBox directory.

HAVE FUN!
The DOSBox Team http://www.dosbox.com

Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>mount x c:/Users/Peanut/AppData/Roaming/Code/User/globalStorage/kaixa.masm-code
Drive X is mounted as local directory c:/Users/Peanut/AppData/Roaming/Code/User/globalStorage/kaixa.masm-code\

Z:\>set PATH=Z:\;x:\;

Z:\>mount c "c:\Users\Peanut\Desktop\DEMO"
Drive C is mounted as local directory c:\Users\Peanut\Desktop\DEMO\

Z:\>c:\

C:\>

C:\>stu
4 4 2 2 3
```

屏幕输出分别为 $[0, 60)$, $[60, 70)$, $[70, 80)$, $[80, 90)$, $[90, 100]$ 区间段的人数

由题可知, $[0, 60)$ 共有 4 人, 分别为 25, 0

$[60, 70)$ 共有 4 人, 分别为 63, 69, 60, 61

$[70, 80)$ 共有 2 人, 分别为 70, 77

$[80, 90)$ 共有 2 人, 分别为 89, 89

$[90, 100]$ 共有 3 人, 分别为 95, 90, 100

所以, 程序运行结果正确

4) 其他要求 (在这里写清题目中其他要求的具体内容)

无

编程题 2:

用汇编语言编写程序, 计算 100 以内所有包含 6 的数字和 6 的倍数的数字的和, 结果显示在屏幕上。

请在答题时, 按照以下格式填写内容:

1、程序源代码（包含必要的注释信息）

2、程序运行截图

1) 程序源代码

```
; 将 data 段中的数据以十进制的形式显示出来
; 首先计算 100 以内所有包含 6 的数字和 6 的倍数的数字的和,
; 以 bx 寄存器 作为累加器,最后最终结果放入 ax 寄存器 中
; 调用 dtoc, 将 ax 转化十进制数的 ASCII 码, 放入 buffer
; 调用 show_str, 将结果以绿色字体输出
assume cs:code, ds:data, ss:stack
data segment
    ;保存转换后的 ASCII
    buffer dw 10 dup (0)
data ends
stack segment
    dw 256 dup (0)
stack ends
code segment
    start:
        ; 初始化
        mov bx,data
        mov ds,bx
        mov bx,stack
        mov ss,bx
        mov si,0
        mov cx,100
        mov bx,0                ;累加器
        mov dl,6                ;dl 存放除数 6(因为 0-100 作为被除数
只要 16 位, 所以除数为 8 位)
        mov dh,10                ;dh 存放除数 10, 用以试探个位、十位
是否有 6
        ; 计算 100 以内所有包含 6 的数字和 6 的倍数的数字的和
    cal_ax:
        mov ax,cx
        div dl                    ;div 指令中: 被除数隐藏在 ax 中, 除数
存放位置为寄存器或者内存单元, 此处为 dl
        cmp ah, 0                ;余数存放在 ah 中
        je jia1                  ;余数=0(为 6 倍数)则跳转到 jia1

        ;不为 6 倍数但是个位有 6
        mov ax,cx
        div dh
        cmp ah,6
```



```

        Je    jia1
;不为 6 倍数，个位无 6 但十位有 6
        mov   ax,cx
        div   dh
        CMP   al,6
        JNE   jia2                ;不相等(个位也无 6)则跳转到 jia2
jia1:
        add   bx,cx
jia2:
        loop  cal_ax

        mov   ax,bx                ;结果最后放入 ax
        mov   dx,0
        call  dtoc                ; 调用 dtoc，将 ax 转化十进制数的
ASCII 码，放入 buffer
        mov   dh,8
        mov   dl,3
        mov   cl,2
        mov   si,0
        call  show_str

        mov   ax,4c00h
        int   21h

dtoc:
; 逐次取余数，并入栈，最后再倒序拿出来，在 buffer 最后放 0
        push  ax
        push  di
        push  cx
        push  dx
        push  si
        mov   di,0                ; 记录入栈次数,就是有多少位数
dtoc_s1:
        mov   cx,10
        mov   dx,0
        div   cx

        mov   cx,ax                ; 如果商为 0，则求值完成
        jcxz  dtoc_s2

        add   dx,30h
        push  dx                ; 求得的 ASCII 码入栈
        inc   di

```

```

        jmp dtoc_s1
dtoc_s2:
        add dx,30h           ;最后一次也要记录
        push dx
        inc di
        mov cx,di           ; cx 为转化后的字符串长度
dtoc_s3:
        pop ax
        mov [si],al         ;ASCII 码只占用了低 8 位
        inc si
        loop dtoc_s3        ; 将 ASCII 码出栈
        mov al,0
        mov [si],al         ; 最后一位放 0 (虽然本来初始化的时
候就已经是 0 了)
        pop si
        pop dx
        pop cx
        pop di
        pop ax
        ret

show_str:
        push cx             ;保存用到的寄存器
        push dx
        mov ax,0b800h
        mov es,ax           ; 用 es 储存显存段地址
; 计算行偏移
        sub dh,10           ; dh-1
        mov bl,10
        mov al,dh
        mul bl              ; 计算出起始行位置, 存在 ax 中
        mov bl,16
        mul bl
        mov bp,ax           ; 行地址暂存在 bp 中
; 计算列偏移
        sub dl,1
        mov al,2
        mul dl
        add ax,bp
        mov di,ax           ; 显示偏移地址放到 di 中
        mov bp,0            ; 清空 bp
        mov bl,cl           ; 字体颜色放到 bl 中

```

```

show:
    mov     al,ds:[si]                ; 把字符串中的一个字节放到 al
    mov     ah,0
    mov     cx,ax                    ; 把每个数据都放到 cx 中检验是否为 0

    jczx    ok

    mov     byte ptr es:[di],al      ; 显示字符
    mov     byte ptr es:[di+1],bl    ; 显示颜色
    add     di,2                     ; 显存指针每次+2
    inc     si                       ; 字符串指针每次+1
    jmp     show

ok:
    pop     dx
    pop     cx
    ret

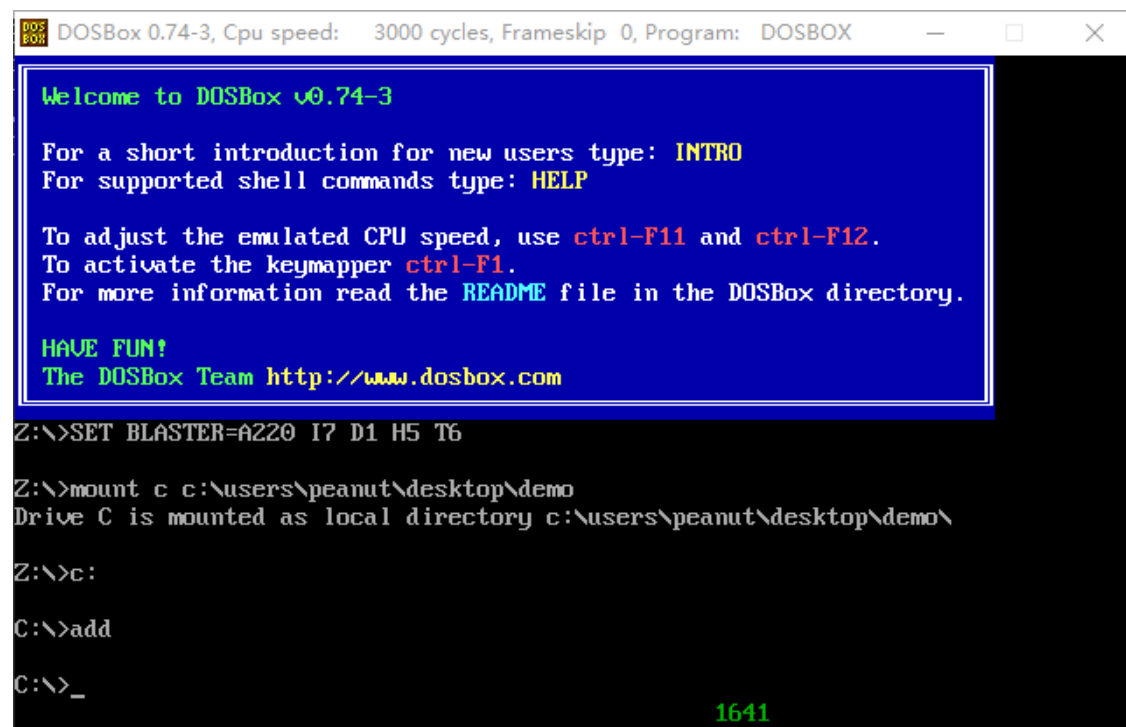
code ends
end start

```

2) 程序源码的屏幕截图



3) 程序运行结果的屏幕截图



The screenshot shows a DOSBox 0.74-3 window. The title bar reads "DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX". The main window has a blue background with white text. It displays a welcome message: "Welcome to DOSBox v0.74-3", followed by instructions for new users to type "INTRO" for an introduction and "HELP" for supported shell commands. It also mentions adjusting CPU speed with "ctrl-F11" and "ctrl-F12", activating the keymapper with "ctrl-F1", and reading the "README" file. The message "HAVE FUN!" and "The DOSBox Team http://www.dosbox.com" are also shown. Below the message box, the command prompt shows the following commands and output: "Z:\>SET BLASTER=A220 I7 D1 H5 T6", "Z:\>mount c c:\users\peanut\desktop\demo", "Drive C is mounted as local directory c:\users\peanut\desktop\demo\", "Z:\>c:", "C:\>add", "C:\>_", and a green number "1641" at the bottom right.

4) 其他要求（在这里写清题目中其他要求的具体内容）

无

3、附加题

1) 题目

编写程序，实现以下功能：

从键盘输入一个年份，例如 2006，编写程序，判断该年份是否为闰年，如果是闰年则输出 “This is a leap year!”，如果不是，则在屏幕上显示 “This is not a leap year!”，并找出距离所输入年份最近的闰年。

同时也要注意：若输入的年份有误，例如 12d3，则输出 “There is a problem with the input!”

2) 考核的知识点

1. 键盘输入。能够利用 DOS 的 21H 中断类型的 10 号键盘功能调用，并从键盘接收到的表示年份的数据字符串按字节存入缓存区。
2. 屏幕输出。能够按照输出要求相关信息。
3. 字符串与整数之间的转换。能够将字符串中的字符逐个转换为对应的十

进制数值

4. 子程序。
5. 循环。
6. 熟练掌握汇编语句，来判断是否为闰年。这里使用到的包括算术运算语句和逻辑运算语句两类，需要在编程时熟悉各语句的用法。

3) 标准答案或程序源代码（包含必要的注释）

```
data segment ;定义数据段
    infon db 0dh,0ah,'Please input a year: $'
    Y      db 0dh,0ah,'This is a leap year! $'
    N      db 0dh,0ah,'This is not a leap year! $'
    w      dw 0
    buf     db 8
           db ?
           db 8 dup(?)
    e      db 0dh,0ah,'There is a problem with the input! $'
    isl     db ' is The most recent leap year! $'
    outbuf  db 200 dup(0)
data ends

stack segment stack
    db 200 dup(0)
stack ends

code segment
    assume ds:data,ss:stack,cs:code
start:
    ;初始化
    mov     ax,data
    mov     ds,ax

    lea     dx,infon ;在屏幕上显示提示信息
    mov     ah,9
    int     21h

    lea     dx,buf ;从键盘输入年份字符串
    mov     ah,10
    int     21h

    mov     cl,[buf+1] ;cl=实际输入的字符数（不包括回车）
```

```

        lea    di,buf+2                ;将年份的地址传给寄存器
        call   datacate

        call   ifyears
        jc     a1                      ;如果判断是闰年就显示 Y, 不
是闰年就显示 N
        lea    dx,n
        mov     ah,9
        int     21h
        call   nearyear

a1:      jmp     exit
        lea     dx,y
        mov     ah,9
        int     21h
        jmp     exit
a2:      lea     dx,e
        mov     ah,9
        int     21h

exit:    mov     ah,4ch
        int     21h

nearyear proc near
        mov     ax,w
        inc     ax
        mov     w,0
        mov     w,ax
        mov     cx,8
loop1:   call    ifyears
        jc     sh
        loop   loop1
sh:
        mov     ax,w
        call    putInt                ;子函数调用
        lea     dx,is1
        mov     ah,9
        int     21h
        ret
nearyear endp

datacate proc near                    ;字符转换

```

```

        push    cx                                ;
        dec     cx
        lea     si,buf+2
tt1:    inc     si
        loop    tt1

        pop     cx                                ;把字符指向最后一个数字
        mov     dh,30h                            ;转换成 ASCII 码,需要减 30h
        mov     bl,10                            ;转为 10 进制
        mov     ax,1
l1:     push    ax
        sub     byte ptr [si],dh                  ;把输入的年份字符减 30H
        cmp     byte ptr [si],9
        jg      a2                                ;有符号大于则跳转
        cmp     byte ptr [si],0
        jl      a2                                ;有符号小于则跳转
        mul     byte ptr [si]                    ;
        add     w,ax                              ;程序把换算成的年份数放在
w 处
        pop     ax
        mul     bl                                ;ax 进位
        dec     si
        loop    l1
        ret

datacate endp

ifyears proc near                                ;判断 w 处的年分数是否为闰
年
        push    ax
        push    bx
        push    cx
        push    dx
        mov     ax,w                            ;将年份弄到 cx 寄存器
        mov     cx,ax
        mov     dx,0
        mov     bx,4
        div     bx
        cmp     dx,0                            ;DX 中放余数
        jnz     lab1                            ;不被 4 整除
        mov     ax,cx
        mov     bx,100
        div     bx
        cmp     dx,0

```

```

        jnz     lab2                ;被 4 整除但不被 100 整除
        mov     ax,cx
        mov     bx,400             ;被 400 整除
        div     bx
        cmp     dx,0
        jz      lab2
lab1:    clc                        ;设置 CF 标志位为 0
        jmp     lab3
lab2:    stc                        ;设置 CF 标志位为 1
lab3:    pop     dx
        pop     cx
        pop     bx
        pop     ax
        ret
ifyears endp

putInt:
        push    si
        push    dx
        mov     dx,0
        mov     si,offset outbuf
        call    dtoc
        mov     dx,offset outbuf
        mov     ah,9
        int     21h
        pop     dx
        pop     si
        ret

        dtoc:                        ; 逐次取余数，并入栈，最后
再倒序拿出来，在 buffer 最后放 0
        push    ax
        push    di
        push    cx
        push    dx
        push    si
        mov     di,0                ; 记录入栈次数
dtoc_s1:
        mov     cx,10
        mov     dx,0
        div     cx

        mov     cx,ax                ; 如果商为 0，则求值完成

```



```

        jcxz    dtoc_s2
        add     dx,30h
        push    dx                ; 求得的 ASCII 码入栈
        inc     di
        jmp     dtoc_s1

dtoc_s2:
        add     dx,30h
        push    dx
        inc     di
        mov     cx,di            ; cx 为转化后的字符串长度

dtoc_s3:
        pop     ax
        mov     [si],al
        inc     si
        loop    dtoc_s3         ; 将 ASCII 码出栈
        mov     al,"$"
        mov     [si],al        ; 最后一位放$(虽然本来初始
                                化的时候就已经是 0 了)
        pop     si
        pop     dx
        pop     cx
        pop     di
        pop     ax
        ret

code ends
end start

```

4) 程序运行截图（如果是编程题）

```

DOSBox 0.74-3, Cpu speed: 3000 cycles/frameskip: 0, Program: DOSBOX
To activate the keymapper ctrl-F1.
For more information read the README file in the DOSBox directory.

HAVE FUN!
The DOSBox Team http://www.dosbox.com

Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>mount x c:/Users/Peanut/AppData/Roaming/Code/User/globalStorage/kaixa.masm-code
Drive X is mounted as local directory c:/Users/Peanut/AppData/Roaming/Code/User/globalStorage/kaixa.masm-code\

Z:\>set PATH=Z:\;x:\;

Z:\>mount c "c:\Users\Peanut\Desktop\DEMO"
Drive C is mounted as local directory c:\Users\Peanut\Desktop\DEMO\

Z:\>c:\>
C:\>year

Please input a year: 1999
This is not a leap year! 2000 is The most recent leap year!

```

```
C:\>year  
Please input a year: 1999  
This is not a leap year! 2000 is The most recent leap year!  
C:\>year  
Please input a year: 2000  
This is a leap year!
```

```
C:\>year  
Please input a year: 222w  
There is a problem with the input!  
C:\>
```

实验思考与分析：

经过本次预习实验，我收获颇多。

首先同时通过自学王爽《汇编语言》，了解了基本的汇编语句用法，掌握了基本的调试方法，在第一个实验中，我深入了解了汇编判断语句的使用，成功写出了正确的汇编代码，在我做的第一个实验（求和）中，让我收获的是答案的屏幕输出：在这里，我参考了王爽的《汇编语言》，利用了其中提到的“显示缓冲区”，即内存地址空间中 B80000H-BFFFFH 共 32KB 的空间是 80×25 彩色字符模式的显示缓冲区，向这个空间写入数据，写入的内容将立即出现在显示器上。了解之后，我将结果转换为 ASCII 的字符串，完成了答案结果的输出。