

汇编语言程序设计

习题课3

助教: 李灯举(hello@lidengju.com)

大纲

- ▶ 第八次作业
- ▶ 第九次作业

对每次作业分为「知识点」和「错题讲解」

第八次作业—知识点

- ▶ 数值运算程序设计
- ▶ 二进制乘除法
 - ▶ MUL / IMUL指令 (单操作数, 隐含使用AL、AX、EAX)
 - ▶ DIV / IDIV 指令 (商和余数, 隐含使用AX、DX:AX、EDX:EAX)
- ▶ BCD码加减法调整指令
 - ▶ 非组合型BCD码 (高位0填充, 一位BCD码占用一个字节)
 - ▶ AAA / AAS
 - ▶ 组合型BCD码 (两位BCD码占用一个字节)
 - ▶ DAA / DAS

第八次作业—题目6

- ▶ 在UnPKBCD中，定义有20个两位非组合型BCD码整数，编写一个计算他们和的程序。假定最终和小于999，且存放在SUM开始的单元中。
- ▶ 例如，数89,34 在UnPKBCD中定义成：09,08,04,03。根据此约定自行定义其余数据。

```
Data Segment
    len equ 20    ; 20个BCD码
    BCD_ARRAY DW len dup(?)    ; 字型BDC数组
    BCD_SUM DB 3 dup(0)    ; 佰, 拾, 个位
Data Ends

Stack0 Segment Stack
    dw 20H dup(?)
Stack0 Ends
```

第八次作业—题目6

Code Segment

Assume CS:code, SS:stack0, DS:Data

Main:

```
MOV    AX, Data
MOV    DS, AX
MOV    CX, len        ; 循环次数
MOV    BX, 0          ; 加数指针
```

Main:

```
MOV    AX, Data
MOV    DS, AX
MOV    CX, len        ; 循环次数
MOV    BX, 0          ; 加数指针
```

NEXT:

```
MOV    AL, BCD_SUM+2   ; 个位
ADD    AL, Byte PTR BCD_ARRAY[BX+0]
AAA                                ; 个位调整, 可能产生CF
MOV    BCD_SUM+2, AL    ; 保存字的和个位
MOV    AL, BCD_SUM+1    ; 拾位
ADC    AL, Byte ptr BCD_ARRAY[BX+1] ; 带进位加法
AAA                                ; 拾位调整
MOV    BCD_SUM+1, AL
ADC    BCD_SUM, 0        ; CF加到百位不调整
INC    BX                ; 移动指针
INC    BX
LOOP   NEXT
MOV    AH, 4CH
INT    21H
```

Code Ends

End Main

第八次作业—题目8

- ▶ 设有两个由6个组合型BCD码组成的多精读数，试编写程序计算他们的和

```
Data Segment
    N equ 6
    BCD_ARRAY1 DB N dup(?)
    BCD_ARRAY2 DB N dup(?)
    BCD_Sum     DB N dup(?), 0 ;多一个最高位
Data Ends

Stack0 Segment Stack
    DW 20H dup(?)
Stack0 Ends
```


第八次作业—题目8

```
Code Segment
    Assume CS: Code, SS: Stack0, DS: Data
Main:
    MOV     AX, Data
    MOV     DS, AX
    MOV     CX, N                ;循环次数为个数
    XOR     BX, BX              ;清零, BX为数组指针
NEXT:
    MOV     AL, BCD_ARRAY1[BX]
    ADC     AL, BCD_ARRAY2[BX]  ;带进位的加法
    DAA                                ;组合BCD加法调整指令
    MOV     BCD_Sum[BX], AL    ;保存到目标数组
    INC     BX
    Loop    NEXT
    ADC     BCD_Sum[BX], 0      ;最高位根据CF累加
    MOV     AH, 4CH
    INT     21H
Code Ends
End Main
```

第九次作业—知识点

- ▶ 非数值处理程序设计
- ▶ 串操作指令 [**LABEL:**][Repeat Prefix] 串操作指令
 - ▶ LODS: 取串
 - ▶ STOS: 存串
 - ▶ MOVS: 串传送
 - ▶ CMPS: 串比较
 - ▶ SCAS: 串搜索
 - ▶ INS: 从口地址读取串
 - ▶ OUTS: 写串数据到口地址
- ▶ 串操作指令的应用

第九次作业—题目4

- ▶ 编写将串中的空隔（20H）删除的程序

```
Data Segment
    len equ 20H
    maxlen db len ;缓冲区最大字符数
    actlen db ?
    String db len dup(?), '$' ;源串
    Stringtemp db len+1 dup(?) ;
    bkey db ' ' ;空白符
    newline db 0A, 0D, '$' ;换行操作
Data ends

;堆栈段定义略

Code Segment
    ASSUME cs: code, ss:略, ds: data
```

第九次作业—题目4

```
Code Segment
    ASSUME cs: code, ss: 略, ds: data
First:
    mov AX, data
    mov DS, AX
    mov ES, AX ;使用串指令必须初始化
    mov DX, offset maxlen ;缓冲区最大字符数
    mov AH, 0aH ;键盘输入到缓冲区
    INT 21H
    lea SI, String ;源串的有效地址
    lea DI, Stringtemp
    mov CX, 0 ;CX清零
    mov CL, actlen ;循环次数
```

```
NEXT:
```

```
    lodsb ;从源串读取一个字节放入AL
    cmp AL, bkey ;判断是否为空白符
    JZ SpaceASCII ;为空则不存
    stosb ;否则存
```

```
SpaceASCII:
```

```
    Loop NEXT ;直到遍历完串
    mov byte ptr [DI], '$' ;字符串用'$'结束
    mov AH, 09H ;显示字符串功能号
    mov DX, offset newline ;换行
    INT 21H
    mov DX, offset stringtemp ;输出目的字符串
    int 21H
    mov ah, 4CH
    INT 21h
```

```
code Ends
```

```
End first
```

第九次作业—题目5

- ▶ 试编写一个程序，在一个指定的字符前插入空隔（20H）的程序

```
Data Segment
Space_ASC equ 20H ;空白字符
len equ 40H ;串长度
actlen db len ;循环次数
Str1 db len dup(?) ;源串
Str2 db len*2 dup(?), '$' ;目的串，长度是源的两倍
Bkey db ? ;指定字符
Data Ends

;堆栈段略

Code Segment

    Assume CS: code, SS: 略, ds: Data
```


第九次作业—题目5

First:

```
mov  AX, Data
mov  DS, AX
mov  ES, AX
lea  SI, Str1 ;取源串的有效地址
mov  DI, offset Str2 ;目的串
mov  CX, 0 ;CX清零
mov  CL, len ;循环次数
CLD ;DF清零, 正向
```

NEXT:

```
LODSB
cmp  AL, Bkey
JNZ  No_char_in ;不为零则跳转存储当前字符
mov  byte ptr[DI], Space_ASC
INC  DI ;数组指针移动
```

NO_char_in:

```
STOSB ;存储字符
Loop NEXT
mov  AH, 4CH
INT  21H
```

Code Ends

End First

THANKS A LOT.

HAPPY TERM END