String Manipulation program
Assembly Language

Dept. of CSIE

Fu Jen Catholic University,

周賜福

Dec 14th, 2018

String Library DEMO program

```
; String Library Demo           (StringDemo.asm)

; This program demonstrates the string-handling procedures in
; the book's link library.

INCLUDE Irvine32.inc

.data
string_1 BYTE "abcde////",0
string_2 BYTE "ABCDE",0
msg0     BYTE "string_1 in upper case: ",0
msg1     BYTE "string1 and string2 are equal",0
msg2     BYTE "string_1 is less than string_2",0
msg3     BYTE "string_2 is less than string_1",0
msg4     BYTE "Length of string_2 is ",0
msg5     BYTE "string_1 after trimming: ",0
```

String Library DEMO program

```
.code
main PROC

        call    trim_string
        call    upper_case
        call    compare_strings
        call    print_length

        exit
main ENDP
```

Procedure 1 trim string

```
trim_string PROC
; Remove trailing characters from string_1.

        INVOKE Str_trim, ADDR string_1,'/'
        mov       edx,OFFSET msg5
        call    WriteString
        mov       edx,OFFSET string_1
        call    WriteString
        call    Crlf

        ret
trim_string ENDP
```

Procedure 2 upper case

```
upper_case PROC
; Convert string_1 to upper case.

        mov       edx,OFFSET msg0
        call     WriteString
        INVOKE  Str_ucase, ADDR string_1
        mov       edx,OFFSET string_1
        call     WriteString
        call     Crlf

        ret
upper_case ENDP
```

Procedure 3 Compare string

```
compare_strings PROC
; Compare string_1 to string_2.

      INVOKE Str_compare, ADDR string_1, ADDR string_2
      .IF ZERO?
      mov     edx,OFFSET msg1
      .ELSEIF CARRY?
      mov     edx,OFFSET msg2     ; string 1 is less than...
      .ELSE
      mov     edx,OFFSET msg3     ; string 2 is less than...
      .ENDIF
      call    WriteString
      call    Crlf

      ret
compare_strings  ENDP
```

Procedure 4 Print Length 5/5

```
print_length PROC
; Display the length of string_2.

        mov      edx,OFFSET msg4
        call     WriteString
        INVOKE  Str_length, ADDR string_2
        call     WriteDec
        call     Crlf

        ret
print_length ENDP

END main
```

Second program

Another Program

1/2

```
; Two-Dimensional Table              (Table.asm)
; Demonstration of Base-Index mode with a
; two-dimensional table.
INCLUDE Irvine32.inc
.data
tableB  BYTE  10h,  20h,  30h,  40h,  50h
        BYTE  60h,  70h,  80h,  90h, 0A0h
        BYTE  0B0h, 0C0h, 0D0h, 0E0h, 0F0h
RowSize = 5
.code
main PROC
; Demonstrate Base-Index mode:
   mov  ebx,OFFSET tableB
   add  ebx,RowSize
   mov  esi,2                                  ; column number
   mov  al,[ebx + esi]                ; AL = 80h
```

Another Program

2/2

```
; Calculate sum of row 1:

        RowNum = 1
        mov  ecx,RowSize
        mov  ebx,OFFSET tableB
        add  ebx,(RowSize * RowNum)        ; move to row 1
        mov  esi,0                                 ; beginning of row
        mov  ax,0                                  ; zero the sum
        mov  dx,0                                  ; holds each value
L1:
        mov   dl,[ebx + esi]           ; get a byte
        add   ax,dx                                ; add to accumulator
        inc   esi
        loopd L1
                                       ; AX = 280h, the sum

        exit
main ENDP
END main
```

Third program

Third Program

1/3

```
; Trim Trailing Characters          (Trim.asm)

; Test the Trim procedure. Trim removes trailing all
; occurrences of a selected character from the end of
; a string.
INCLUDE Irvine32.inc
Str_trim PROTO,
        pString:PTR BYTE,              ; points to string
        char:BYTE                            ; character to remove

Str_length PROTO,
        pString:PTR BYTE               ; pointer to string

ShowString PROTO,
        pString:PTR BYTE
```

Third Program

2/3

```
.data
; Test data:
string_1 BYTE 0                 ; case 1
string_2 BYTE "#",0             ; case 2
string_3 BYTE "Hello###",0      ; case 3
string_4 BYTE "Hello",0         ; case 4
string_5 BYTE "H#",0            ; case 5
string_6 BYTE "#H",0            ; case 6
```

```
.code
main PROC
        call Clrscr
        INVOKE Str_trim,ADDR string_1,'#'
        INVOKE ShowString,ADDR string_1
        INVOKE Str_trim,ADDR string_2,'#'
        INVOKE ShowString,ADDR string_2
        INVOKE Str_trim,ADDR string_3,'#'
        INVOKE ShowString,ADDR string_3
        INVOKE Str_trim,ADDR string_4,'#'
        INVOKE ShowString,ADDR string_4
        INVOKE Str_trim,ADDR string_5,'#'
        INVOKE ShowString,ADDR string_5
        INVOKE Str_trim,ADDR string_6,'#'
        INVOKE ShowString,ADDR string_6
        exit
main ENDP
```

3/3

```
;------------------------------------------------
ShowString PROC USES edx, pString:PTR BYTE
; Display a string surrounded by brackets.
;------------------------------------------------
.data
lbracket BYTE "[",0
rbracket BYTE "]",0
.code
        mov  edx,OFFSET lbracket
        call WriteString
        mov  edx,pString
        call WriteString
        mov  edx,OFFSET rbracket
        call WriteString
        call Crlf
        ret
ShowString ENDP
END main
```

Practice these string programs for Exam