Andrew Tran

Lab Experience Thirteen

**Objectives:**

1. Using the string primitive instructions
2. Manipulating strings.

**What you must do:**

1. Do problem #2 on page 386.
2. Do problem #4 on page 387.

Copy and paste each program into a word document along with screen shots showing the execution of your program.

**What to hand in:**

1. Printouts of the above programs
2. Printout of the word document.
3. Compress all of the above programs into a single file (.zip) using your name as part of the file name. i.e. timwrennlab13.zip.
4. Place the zipped file into the Lab Thirteen D2L Drop Box Folder.

**Due Date:**

As indicated on the drop box folder.

Problem 1:

INCLUDE Irvine32.inc

Str\_concat PROTO, source:PTR BYTE, target : PTR BYTE

.data

targetStr BYTE "ABCDE", 10 DUP(0)

sourceStr BYTE "FGH", 0

.code

main PROC

mov edx, OFFSET targetStr

call WriteString

call Crlf

call WaitMsg

mov eax, LENGTHOF sourceStr

INVOKE Str\_concat, ADDR sourceStr, ADDR targetStr

mov edx, OFFSET targetStr

call WriteString

call Crlf

exit

main ENDP

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

Str\_concat PROC USES eax ecx esi edi, source:PTR BYTE, target : PTR BYTE

mov ecx, eax ; LENGTHOF source

mov esi, source

mov edi, target

add edi, 5

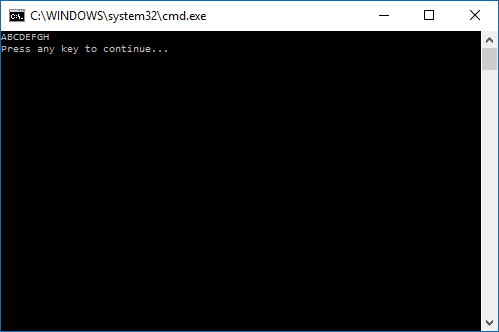
cld ; forward

rep movsb ; copy

ret

Str\_concat ENDP

END main



Problem 2:

INCLUDE Irvine32.inc

Str\_find PROTO, sourcePtr:PTR BYTE, targetPtr:PTR BYTE

.data

source BYTE "Andrew Tran Lab 12", 0

target BYTE "rew", 0

lengthA DWORD ?

response BYTE "Target was found."

.code

main PROC

mov edx, OFFSET source

call WriteString

call crlf

mov ecx, (SIZEOF target)

mov edx, OFFSET target

call WriteString

call crlf

INVOKE Str\_find, ADDR source,ADDR target

jz quit

Found:

mov edx, OFFSET response

call WriteString

call crlf

quit:

main ENDP

exit

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

Str\_find PROC, sourcePtr:PTR BYTE, targetPtr : PTR BYTE

mov eax, sourcePtr

mov ebx, targetPtr

mov esi, targetPtr ;beginning of the string

xor ecx, ecx ;ecx = 0

L0:

cmp BYTE PTR[esi], 0

je L1 ;true

inc esi

inc ecx

jmp L0; FALSE:Jump to L0 ;false

L1 :

mov edx, ecx

L2:

mov esi, eax ;cmpsb setup

mov edi, ebx ;cmpsb setup

repe cmpsb

jz Found

mov ecx, edx

cmp BYTE PTR[eax + ecx], 0

jz Not\_Found

inc eax

jmp L2

Not\_Found:

or eax, 1 ;reset zero flag

ret ;return failure

Found:

mov eax, ebx

call dumpregs ;show eax contents

mov lengthA, edx

sub esi, lengthA

ret

Str\_find ENDP

END main

