Assembly Language Programming Practice Session

Dept. of CSIE

Fu Jen Catholic University,

Hsin Chuang, 24205. Taipei.

Dec 21st, 2018

9.3 Str_remove program

Write a procedure named Str_remove that removes n characters from a string. Pass a pointer to the position in the string where the characters are to be removed. Pass an integer specifying the number of characters to remove. The following code, for example shows how to remove “xxxx” from target:

```
.data
target BYTE “abcxxxxdefghijklmop”,0
.code
INVOKE str_remove, ADDR [target+3], 4
```

Let us write the program as follows:

```
 Comment !
Description: Write a procedure named Str_remove that removes n characters
from a string. Pass a pointer to the position in the string where the
characters are to be removed. Pass an integer specifying the number of
characters to remove.
!
INCLUDE Irvine32.inc
Str_remove PROTO,
pStart:PTR BYTE,
nChars:DWORD
.data
target BYTE "abcXXXXdefghijklmnop",0
```

```
.code
main PROC
INVOKE Str_remove, ADDR [target+3], 4
mov edx,OFFSET target
call WriteString
call Crlf

INVOKE Str_remove, ADDR [target+2], 19; nChars too large
mov edx,OFFSET target
call WriteString
call Crlf

exit
main ENDP
```

```
;-----------------------------------------------------
Str_remove PROC,
pStart:PTR BYTE,; points to first character to delete
nChars:DWORD; number of characters to delete
;
; Removes a block of characters from a string. The
; algorithm involves skipping over the positions that
; will be deleted, and copying the remaining characters
; backward. Each copied character overwrites one of the
; deleted positions. If nChars is too large, all remaining
; characters in the string will be removed.
;-----------------------------------------------------
INVOKE Str_length, pStart
mov  ecx,eax   ; ECX = length of string
```

```
.IF nChars <= ecx                ; check range of nChars
  sub ecx,nChars                 ; set counter for REP prefix
.ENDIF

mov esi, pStart          ; points to string
add esi, nChars          ; points to first character to copy
mov edi, pStart          ; points to destination position

cld                ; clear direction flag (forward)
rep movsb          ; do the move
mov BYTE PTR [edi],0             ; insert new null byte

Exit_proc:
ret
Str_remove ENDP
END main
```

Another Example of String program to practice

9.4 Write a procedure named Str_find that searches for the first matching occurrence of a source string inside a target string and returns the matching position. The input parameters should be a pointer to the source string and a pointer to the target string. If a match is found, the procedure sets the Zero flag and EAX points to the matching position in the target string. Otherwise, the Zero flag is clear and EAX is undefined.

The following code, for example searches for "ABC" and returns with EAX pointing to the "A" in the target string:

```
.data
target BYTE "123ABC342432",0
source BYTE "ABC",0
pos DWORD ?
.code
INVOKE Str_find, ADDR source, ADDR target
jnz notFound
mov pos, eax
```

```
Comment !
Description: Write a procedure named Str_find that searches for the
first matching occurrence of a source string inside a target string
and returns the matching position. The input parameters should be a
pointer to the source string and a pointer to the target string. If
a match is found, the procedure sets the Zero flag and EAX points to
the matching position in the target string. Otherwise, the Zero
flag is clear.
!
INCLUDE Irvine32.inc
Str_find PROTO, pTarget:PTR BYTE, pSource:PTR BYTE
.data
target BYTE "01ABAAAAAABABCC45ABC9012",0
source BYTE "AAABA",0
str1 BYTE "Source string found at position ",0
str2 BYTE " in Target string (counting from zero).",0Ah,0Ah,0Dh,0
str3 BYTE "Unable to find Source string in Target string.",0Ah,0Ah,0Dh,0
```

```
stop      DWORD ?
lenTarget DWORD ?
lenSource DWORD ?
position  DWORD ?
.code
main PROC

INVOKE Str_find,ADDR target, ADDR source
mov  position,eax
jz   wasfound; ZF=1 indicates string found

mov  edx,OFFSET str3; string not found
call WriteString
jmpquit
```

```
wasfound:                   ; display message
mov  edx,OFFSET str1
call WriteString
mov  eax,position                   ; write position value
call WriteDec
mov  edx,OFFSET str2
call WriteString

quit:
exit

main ENDP
```

```
;-----------------------------------------------------
Str_find PROC, pTarget:PTR BYTE, ;PTR to Target string
               pSource:PTR BYTE  ;PTR to Source string
;
; Searches for the first matching occurrence of a source
; string inside a target string.
; Receives: pointer to the source string and a pointer
; to the target string.
; Returns: If a match is found, ZF=1 and EAX points to
; the offset of the match in the target string.
; IF ZF=0, no match was found.
;-----------------------------------------------------
```

```
INVOKE Str_length,pTarget              ; get length of target
mov  lenTarget,eax
INVOKE Str_length,pSource              ; get length of source
mov  lenSource,eax

mov  edi,OFFSET target          ; point to target
mov  esi,OFFSET source          ; point to source

; Compute place in target to stop search
mov  eax,edi                ; stop = (offset target)
add  eax,lenTarget                  ;      + (length of target)
sub  eax,lenSource                  ;      - (length of source)
inc  eax                            ;      + 1
mov  stop,eax               ; save the stopping position
```

```
; Compare source string to current target
cld
mov  ecx,lenSource     ; length of source string
L1:
pushad
repe cmpsb             ; compare all bytes
popad
je   found             ; if found, exit now

inc  edi; move to next target position
cmp  edi,stop          ; has EDI reached stop position?
jae  notfound          ; yes: exit
jmp  L1; not: continue loop

notfound:              ; string not found
or   eax,1; ZF=0 indicates failure
jmp  done
```

```
found:              ; string found
mov  eax,edi                ; compute position in target of find
sub  eax,pTarget
cmp  eax,eax                    ; ZF=1 indicates success

done:
ret
Str_find ENDP

END main
```

END of the slides of sample programs