

Title: Write 8086 assembly program to input a string and reverse it.

Batch: A2 Roll No.: 16010421063 Experiment No.: 08

Aim: Write an 8086 assembly program to input a string and reverse it.

Resources needed: DOSBOX, TASM

Theory:

String manipulation is one of the most common operations in programming (in any language). Some of the most common string functions are strepy, strlen, streat, strrev, strehr, strstr. strtok etc.

In this program we first input a string using the DOS function AH=0AH, INT 21H.

The string buffer which stores the incoming string has the following format:-

- 1. The first byte indicates the maximum characters which the user is allowed to enter. If more characters are entered they are simply ignored.
- 2. The second character is filled with the actual number of characters entered by the user including the enter key (last character is 0x0d).
- 3. The remaining space holds the actual input string.

We then check for an empty string; if so display an error message and exit.

Otherwise we reverse the string one character at a time and then display both the original and the reversed string.

; Program: Input a string and reverse it.; FileName: expt8.asm

dispstr macro strname mov ah, 09h mov dx, offset strname int 21h endm

crlf macro

mov dl, 0ah mov ah, 02h int 21h mov dl, 0dh mov ah, 02h int 21h endm

.model small

.stack 100h

```
.data
buff
       db 20
                 ;MAX NUMBER OF CHARACTERS ALLOWED (18).
      db?
               ;NUMBER OF CHARACTERS ENTERED BY USER.
       db 18 dup(0) ;CHARACTERS ENTERED BY USER(17)+0x0d
str1
rbuff
              db
                     18 \, dup(0)
              db "Orig String:",'$'
msg1
              db "Rev String:",'$'
msg2
              db "Empty String", '$'
msg3
.code
main:
      mov ax, @data
      mov ds, ax
                     mov es, ax
CAPTURE STRING FROM KEYBOARD.
      mov ah, 0Ah; SERVICE TO CAPTURE STRING FROM KEYBOARD.
      mov dx, offset buff
      int 21h
;CHANGE CHR(13) BY '$'.
      mov si, offset buff + 1; NUMBER OF CHARACTERS ENTERED.
      mov cl, [si]; MOVE LENGTH TO CL.
      mov ch, 0
                 CLEAR CH TO USE CX.
      inc cx; TO REACH CHR(13).
      add si, cx; NOW SI POINTS TO CHR(13).
      mov al, '$'
      mov [ si ], al ;REPLACE CHR(13) BY '$'.
                     crlf
do reverse:
      mov si, offset buff + 1; NUMBER OF CHARACTERS ENTERED.
      mov cl, [si]; MOVE LENGTH TO CL.
      mov ch, 0
                  ;CLEAR CH TO USE CX.
                                     ; Check for empty string
                     cmp cx,0
                     je empty
                     add si, cx
                                          ; Make si point to end of string
                     mov di, offset rbuff
                                          ; Make di point to start of reverse string
nxtchr:
                                   ; Read next character from si from right
                     mov al, [si]
                     mov [di], al
                                   ; Store in rbuff
```

; update pointers

dec si

inc di

loop nxtchr; loop decrements cx and checks for 0,

; If not 0 repeat the loop

mov al, '\$' ; Add '\$' to indicate end of string

mov [di], al

; Now display both strings

dispstr msg1 ; Display original string

dispstr str1

crlf

dispstr msg2 ; Display reversed string

dispstr rbuff jmp over

empty:

dispstr msg3

crlf

over: mov ah, 4ch ; program over, call exit() function

int 21h

end

Procedure:

- 1. Download and install DOSBox and TASM in C drive
- 2. Type the programs given in this file and save them as .asm files.
- 3. Run DosBox
- 4. In the DosBox window, compile and run the program:
 - > tasm /l expt8.asm
 - > tlink expt8.obj
 - > expt8.exe
- 5. Verify the desired output

Observations and Results:

Both the programs should display the desired outputs in the DOSBOX window. Paste the snapshots here.

C:\>exp7.exe Hello World Hello World C:\>tasm /l reverse.asm Turbo Assembler Version 2.51 Copyright (c) 1988, 1991 Borland International Assembling file: reverse.asm Error messages: None Warning messages: None Passes: Remaining memory: 484k C:∖>tlink re∪erse.obj Turbo Link Version 4.0 Copyright (c) 1991 Borland International C:\>reverse.exe Empty String C:\>reverse.exe Hello Orig String:Hello Re∨ String:olleH

Outcomes:

CO 4. Understand the basic concepts of memory elements.

CO 5. Understand the fundamental concepts of microprocessors.

Conclusion:

We learned how to reverse a string using assembly language

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of faculty in-charge with date

References:

Books/ Journals/ Websites:

1. "Microprocessors and Interfacing", by Douglas Hall Tata McGraw Hill 3rd edition.