



Vidyavardhini's College of Engineering & Technology

Department of Artificial Intelligence and Data Science (AI&DS)

Name:	Swarup Satish Kakade
Roll No:	19
Class/Sem:	SE/IV
Experiment No.:	6
Title:	To perform program to reverse the word in string
Date of Performance:	
Date of Submission:	
Marks:	
Sign of Faculty:	



Vidyavardhini's College of Engineering & Technology

Department of Artificial Intelligence and Data Science (AI&DS)

Aim: Assembly Language Program to reverse the word in string.

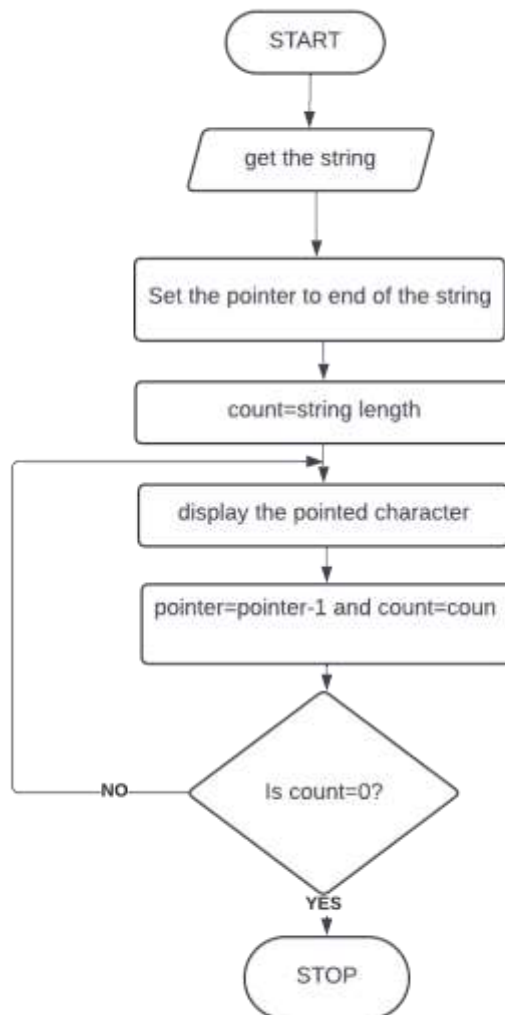
Theory:

This program will read the string entered by the user and then reverse it. Reverse a string is the technique that reverses or changes the order of a given string so that the last character of the string becomes the first character of the string and so on.

Algorithm:

1. Start
2. Initialize the data segment
3. Display the message -1
4. Input the string
5. Display the message 2
6. Take characters count in DI
7. Point to the end character and read it
8. Display the character
9. Decrement the count
10. Repeat until the count is zero
11. To terminate the program using DOS interrupt
 - a. Initialize AH with 4ch
 - b. Call interrupt INT 21h
12. Stop

Flowchart:



Assembly Code:

```
original source code
001
002 ; You may customize this and other start-up templates;
003 ; The location of this template is c:\emu8086\inc\8_con_template.txt
004
005 org 100h
006
007 .data
008 n1 db 10,13,'enter string: $'
009 n2 db 10,13,'your string is: $'
010 buff db 80
011
012 .code
013 lea dx,n1
014 mov ah,09h
015 int 21h
016
017 lea dx,buff
018 mov ah,0ah
019 int 21h
020
021 lea dx,n2
022 mov ah,09h
023 int 21h
024
025 mov ch,00h
026 mov cl,[buff+1]
027 lea bx,buff+1
028 add bx,cx
029
030 L1:
031 mov dx,[bx]
032 mov ah,02h
033 int 21h
034 dec bx
035 loop L1
```



Output:

```
emulator screen (191x63 chars)
enter string: yo
your string is: oy
```

Conclusion:

1. Explain the difference between XLAT and XLATB

Ans. XLAT and XLATB are x86 assembly language instructions used for table lookups, but they have some differences:

a) XLAT:

- a. XLAT is an instruction that performs a lookup in a translation table.
- b. It uses the AL register as an index to access a byte-sized entry in a translation table, typically located in memory.
- c. After the lookup, XLAT replaces the content of AL with the byte value found at the memory address calculated using the sum of the contents of AL and the base address of the translation table.

b) XLATB:

- a. XLATB is essentially the same as XLAT, but it's a legacy alternative mnemonic for the same instruction. The "B" suffix in XLATB stands for "byte."
- b. In terms of functionality, XLATB operates exactly like XLAT and performs the same table lookup using the AL register as an index.

In summary, both XLAT and XLATB instructions perform byte-sized table lookups using the content of the AL register as an index. The difference lies in their naming convention, with XLAT being the more commonly used mnemonic, while XLATB is a legacy alternative. The choice between them is mostly a matter of programmer preference.

2. Explain the instruction LAHF.

Ans. The LAHF instruction, standing for "Load AH from Flags," is an x86 assembly language instruction used to load the contents of the flags register into the AH register. Here's an explanation of the LAHF instruction:

a) Purpose:

- a. LAHF is used to transfer the status of the lower byte of the flags register (EFLAGS) into the AH register.
- b. The flags register contains various status flags that reflect the outcome of arithmetic and logical operations, such as the zero flag (ZF), sign flag (SF), carry flag (CF), and others.

b) Usage:

- a. LAHF is typically used in conjunction with the SAHF instruction (Store AH into Flags), which performs the opposite operation.
- b. Before using LAHF, the programmer often saves the contents of the AH register if it holds important data, as LAHF will overwrite its contents.



Vidyavardhini's College of Engineering & Technology

Department of Artificial Intelligence and Data Science (AI&DS)

- c. After executing LAHF, the AH register contains the lower byte of the flags register, where each bit corresponds to a specific flag's status.

- c) Example:

- a. For instance, if the carry flag (CF) and the zero flag (ZF) are set in the flags register, the value loaded into AH by LAHF would be 00000011 in binary, corresponding to a hexadecimal value of 0x03.

- d) Flags Represented:

- a. The bits in the AH register represent the status of the following flags (from bit 0 to bit 7): CF, reserved, PF, reserved, AF, reserved, ZF, SF.

In summary, the LAHF instruction is used to load the lower byte of the flags register into the AH register, providing access to the status of various processor flags for further manipulation or decision-making in assembly language programs.



Vidyavardhini's College of Engineering & Technology

Department of Artificial Intelligence and Data Science (AI&DS)
