



## Department of Computer Engineering

**CLASS:S.E. COMP**

**SUBJECT: MPL**

**EXPT. NO.: 02**

**Name: Dheeraj Gonchigar**

**Roll No.: 21129**

### **PROBLEM STATEMENT:**

Write an X86/64 ALP to accept a string and display its length.

### **OBJECTIVE:**

To

1. Get familiar with read, write system calls used in assembly.
2. Learn hex to ASCII conversion
3. Learn use of read-write macros and procedures.

### **OUTCOME:**

Student will be able to:

1. Write and execute assembly language program using concept of hex to ascii conversion
2. Check the contents of gdb to see various register contents.
3. Understand the difference between macros and procedures.

### **THEORY :**

Macro is a set of instructions and the programmer can use it anywhere in the program by using its name. It is mainly used to achieve modular programming. So the same set of instructions can be used multiple times whenever required by the help of macro. Wherever macro's identifier is used, it is replaced by the actual defined instructions during compilation thereby no calling and return occurs.

Procedures are also like macro, but they are used for a large set of instructions when a macro is useful for a small set of instructions. It contains a set of instructions which performs a specific task. It contains three main parts i.e Procedure name to identify the procedure, procedure body which contains set of instructions, and RET statement which denotes return statement.

Need of hex to ASCII conversion is because the input string is in ASCII form. In the accumulator, the length gets stored in hexadecimal form. So we need to do hex to ascii conversion so that the displayed string is again in ascii form on the console.



## Department of Computer Engineering

### Algorithm:

1. Start
2. Declare string variable s1 of resb of 100 bytes.
3. Declare length variable of resb to size 2 bytes.
4. Display enter a string message on console with read write macro.
5. Move contents of ax into bx register. ax contains the length of the entered string.
6. Call procedure
7. Mov length into rsi register
8. Mask to get the last digit
9. Use the hex to ascii logic conversion written below
10. Cmp al,09h
11. Jc Label 30:
12. Add al,07h
13. Label 30:
14. Add al,30h
15. Mov [rsi],al
16. Inc rsi
17. Dec byte[count]
18. Jnz up
19. Ret
20. Write msg2 prompt
21. Write length of string stored on screen
22. Exit system calls
23. Stop

### Test Cases:-

Input	Expected Output	Actual Output	Status
Hello	06	06	Success
PICTPUNE	000@	000@	Success
i am dheeraj from pict pune	001C	001C	Success

### CONCLUSION:

Hence from this assignment we have calculated the length of the given input string and implemented hex to ASCII conversion.