

Computer Engineering Department

4

MACHINE PROBLEM

CPO102L

**STRING MANIPULATION**

## Section No. W31

Grade:

Lorenzo Sebastianne C. Magno

***Format &***

***Presentation( 5% ) : \_\_\_\_\_\_\_\_***

***Timeliness (20%) : \_\_\_\_\_\_\_\_***

***Correctness( 15% ) : \_\_\_\_\_\_\_\_***

***Summary &***

***Conclusion (20%) :\_\_\_\_\_\_\_\_***

***Machine Code (40%) :\_\_\_\_\_\_\_\_***

Student’s Signature over Printed Name

January 21, 2016

Date Submitted

Honeylet D. Grimaldo

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Instructor, CPO102L

**OBJECTIVES**

1. To learn how to print strings using service 9 of INT 21h

2. To learn how to process string in assembly language level

3. To apply loop and call & return instruction.

3. To create a program that provides a simple screen output by using the string output service.

**BASIC INFORMATION**

Strings can’t usually fit in a register, string are then placed in the memory and then pass the address of the string in memory of two of the registers, the segment address in DS and offset address in DX.

The string output sends a string of characters to the standard output.

On entry :AH=09h.

DS = segment address of the first character of the string

DX = offset address of the first character of the string.

Service 9 display string of character starting with the first character (address in DS: DX) output, but not including, the character “$” (ASCII24H)

**REQUIREMENTS :**

* 1 personal computer
* Asembler – **TASM.EXE**
* Loader - **TLINK.EXE**

**PROCEDURES**

1. Encode the given program. Assign **filename sam3.asm**

.model small

.stack

.data

x db "RED$"

y db "BLUE$’"

.code

org 100h

start:

main proc

movax,@data

movds,ax

mov ah, 9

lea dx, y

int 21h

call down

mov ah, 9

mov dx, offset x

int 21h

mov ah,4ch

int 21h

mainendp

down proc

mov ah, 2

mov dl,13

int 21h

mov dl,10

int 21h

ret

downendp

end start

2. Write down the output of the given program.



3. Modify **sam3.asm**. The output should be:

( Note: Use the **call/ret**instructions )

RED

RED

RED

RED

BLUE

BLUE

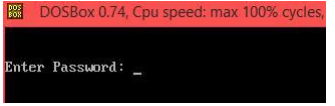
BLUE

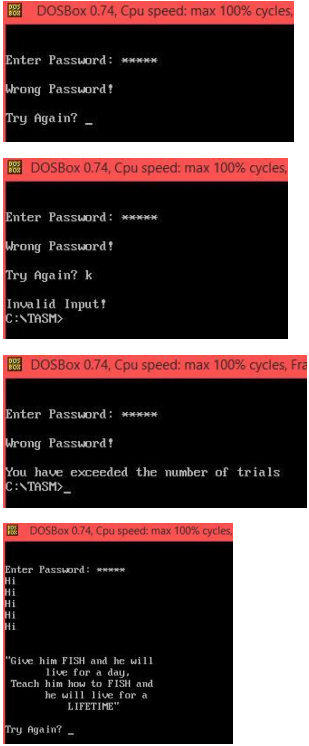
BLUE

**EXERCISES:**

1. 1. Write the required program output from your instructor.

**(Filename: expt3.asm)**







2. Encode the program.

.model small

.stack

.data

pass db 10,10,13,"Enter Password: $"

bye db 10,10,13, "GOODBYE!$",10,10,13

wrongInput db 10,10,13, "Invalid Input!$"

varcount dw 3

ex db 10,10,13, "You have exceeded the number of trials$"

quotes db 10,10,10,13,34, "Give him FISH and he will",10,13," live

for a day,",10,13," Teach him how to FISH and",10,13," he will live for

a",10,13," LIFETIME",34,"$"

tryagain db 10,10,13, "Try Again? $"

hello db 10,13, "Hi$"

wrongPass db 10,10,13, "Wrong Password!$"

. code

org 0100h

start:

mov ax, @data

mov ds, ax

mov ax, 03

int 10h

mov ah, 9

lea dx, pass ;Enter

Password

int 21h

mov ah, 7

int 21h

mov dh, al

mov ah, 2

mov dl, '\*'

int 21h

mov ah, 7

int 21h

mov bl, al

mov ah, 2

mov dl, '\*'

int 21h

mov ah, 7

int 21h

mov bh, al

mov ah, 2

mov dl, '\*'

int 21h

mov ah, 7

int 21h

mov ch, al

mov ah, 2

mov dl, '\*'

int 21h

mov ah, 7

int 21h

mov cl, al

mov ah, 2

mov dl, '\*'

int 21h

jmp comp1

jump:

jmp start

comp1:

cmp dh, 'a'

jne invalid

cmp bl, 'b'

jne invalid

cmp bh, 'c'

jne invalid

cmp ch, 'd'

jne invalid

cmp cl, 'e'

jne invalid

welcome:

mov cx, 5

hii:

mov ah, 9

lea dx, hello

int 21h

loop hii

mov ah, 9

lea dx, quotes

int 21h

try:

mov ah, 9

lea dx, tryagain

int 21h

mov ah, 1

int 21h

mov dh, al

cmp dh, 'y'

je jump

cmp dh, 'Y'

je jump

cmp dh, 'n'

je endd

cmp dh, 'N'

je endd

mov bx, 5

mov ax, @data

mov ds, ax

mov ah, 9

lea dx,

wrongInput

int 21h

mov ah, 4ch

int 21h

endd:

mov ax, @data

mov ds, ax

mov ah, 9

lea dx, bye

int 21h

mov ah, 4ch

int 21h

invalid:

dec bx

mov ax, @data

mov ds, ax

mov ah, 9

lea dx, wrongPass

int 21h

jmp try1

count proc

cnt:

call

enterPass

ret

endp count

enterPass proc

mov ax, 03

int 10h

mov ah, 9

lea dx, pass ;Enter

Password

int 21h

mov ah, 7

int 21h

mov dh, al

mov ah, 2

mov dl, '\*'

int 21h

mov ah, 7

int 21h

mov bl, al

mov ah, 2

mov dl, '\*'

int 21h

mov ah, 7

int 21h

mov bh, al

mov ah, 2

mov dl, '\*'

int 21h

mov ah, 7

int 21h

mov ch, al

mov ah, 2

mov dl, '\*'

int 21h

mov ah, 7

int 21h

mov cl, al

mov ah, 2

mov dl, '\*'

int 21h

jmp comp2

endd1:

jmp endd

comp2:

cmp dh, 'a'

jne

invalid1

cmp bl, 'b'

jne

invalid1

cmp bh, 'c'

jne

invalid1

cmp ch, 'd'

jne

invalid1

cmp cl, 'e'

jne

invalid1

jmp

welcome1

cnt1:

jmp cnt

jump1:

call count

welcome1:

mov ah, 9

lea dx,

quotes

int 21h

try1:

mov ah, 9

lea dx,

tryagain

int 21h

mov ah, 1

int 21h

mov dh, al

cmp dh, 'y'

;mov ah, 9

je jump1

cmp dh,

'Y'

je jump1

cmp dh, 'n'

;mov ah, 9

je endd1

;jne

invalid

cmp dh,

'N'

je endd1

;jne

invalid

mov ax,

@data

mov ds, ax

mov ah, 9

lea dx,

wrongInput

int 21h

mov ah,

4ch

int 21h

jmp cnt1

ret

invalid1:

dec

varcount

mov ax,

@data

mov ds, ax

mov ah, 9

lea dx,

wrongPass

int 21h

equalZero:

cmp

varcount, 1

jne try1

mov ax,

@data

mov ds, ax

mov ah, 9

lea dx, ex

int 21h

mov ah,

4ch

int 21h

endp enterPass

end start

3. Ask your instructor to check your work and save it.

4. What are the requirements that must be satisfied before **INT 21hservice 9** prints the string?

Just like in any other programming language, the int21h service 9 first needs to declare the string in type of DB, like in other declaring it as a string, but here we terminate a string with a dollar sign. Then printing it requires the data segment to be determined.

5. What are the advantages of using procedures?

Procedures help organize your code, can make it easier to debug and avoids duplicating the same code. Meaning a programmer can use one block of code and reuse it so the program will be easier to write and it will have less lines of codes. However, this is best used in large chunks of codes.

6. What is the use of **RET** instruction?

It is paired with the INSTRUCTIONS to properly terminate at the end of the mechanism. .  Issue the ret instruction within the called procedure to resume execution flow at the instruction following the call. In other programming languages, it is called as return.

7. Explain how the **call and ret** instruction works.

The CALL instruction transfers the control to another procedure in code like a pointer, while the RET returns to the instruction after the call.  Issue the ret instruction within the called procedure to resume execution flow at the instruction following the call.

**SUMMARY & CONCLUSION:**

First in the summary and conclusion is the service 9 of int21h, it made our coding easier as we learned how to process the strings, unlike in the first activities wherein we put letters line by line. And also in this program, we learned how to use the loop which is very useful in any language used by a programmer. The CALL and the RET also makes programming easier, these are instructions that will always be used and are always helpful in coding a not-so-simple program.