Term: 412 **Year:** 2019/2020

Course: Comp. Arc. & Assembly Lang. [CCS 3310]

Sections: 1638 - 1640

الجامعة الإسلامية بالمدينة المنورة
ISLAMIC UNIVERSITY OF MADINAH
كلية الحاسب الآلي ونظم المعلومات
Faculty of Computer and Information Systems

--------- Lab Project ---------

Report

Members of Grope

Abdulkader Albai	{372029793}
Mohamed Nasser Hashem	{372029063}

Introduction

Searching is one of the most important operations in many applications. In this project, we implemented the searching operation in Assembly language.

Aim

Find whether a certain word exist in the Video RAM or not. And if it was found then we find what is number of occurrences.

Objectives

- Build a menu with for the user to select from
- After accepting an entry from the user, it will be compared and according to the entry the right selection will be made.
- First option for the user was to select the length of word searched for.
- After the user enters the length of the word he will return to the menu and can choice to enter the word.
- After entering the word, the program will return to the menu if the user choices to find whether the entered word exist in V-RAM the program will start searching from the v-ram index. The program will compare the entered word and the words in the v-ram, if the word is found it will increment a specific counter to count the number of occurrences, and will keep searching until the end of v-ram
- After finishing the search if the word is found it will print "found".
- If the user choices to see how many occurrences. The program will print the number in the counter.
- And the exit choice is always available in the menu for the user

ABET
Computing
Accreditation
Commission

Coding

```
data segment
     word db 9 dup (?)

     error       db "Choose from the list <1 - 5> $"

     S1_length   db "Length <1 - 9> : $"
     S1_number   dw ?

     S2_word     db "Word: $"
     S2_error    db "Enter small letters! $"

     S3_found    db 'Word Found$'
     S3_notfound db 'Word NOT Found$'

     S4_number   db 0ffh dup ('$')
     S4_count    db 0
     S4_counter  db ?
     S4_notfound db 'Word NOT Found $'
     S4_found    db ' Word(s) found $'


     S5          db "Thanks for using our application",0ah,0dh,"$"

     NewLine db 0ah,0dh, "$"
;------------------------------------------------------------------
     mas1 db  "Welcom", 0ah,0dh, "$"
     mas2 db  "choose a service from the following: ",0ah,0dh,"$"
     mas3 db  "press 1 to To Select the length of word searched",0ah,0dh,"$"
     mas4 db  "press 2 to Enter the word ",0ah,0dh,"$"
     mas5 db  "press 3 to Find the entered word in V-RAM",0ah,0dh,"$"
     mas6 db  "press 4 to Find the number of occurrences of the entered word in V-RAM
",0ah,0dh,"$"
     mas7 db  "press 5 to Exit $"

ends

stack segment
    dw 16 dup()
ends
```

```
code segment
Main_prog proc far
  Assume SS:stack, CS:code, DS:data
        Mov AX,data
        Mov DS,AX

            OpenMenu:

            mov dx,offset mas1
            mov ah, 9
            int 21h

            mov dx,offset mas2
            mov ah,9
            int 21h

            mov dx,offset mas3
            mov ah,9
            int 21h

            mov dx,offset mas4
            mov ah,9
            int 21h

            mov dx,offset mas5
            mov ah,9
            int 21h

            mov dx,offset mas6
            mov ah,9
            int 21h

            mov dx,offset mas7
            mov ah,9
            int 21h

            mov dx,offset NewLine
            mov ah,9
            int 21h

      ;----------------------------------
            mov ah, 1
            int 21h
            sub al, 30h
            mov cl, al

            mov dx,offset NewLine
            mov ah,9
            int 21h

            cmp cl, 7
            jae mismatch
            ; we are doing comparisons to choose which service to provide
            cmp cl,1
            je  Service1

            cmp cl,2
            je  Service2

            cmp cl,3
            je  Service3

            cmp cl,4
            je  Service4

            cmp cl,5
            je exit

            mismatch:

                mov dx, offset NewLine
                mov ah, 9
                int 21h

                mov dx, offset error
                mov ah, 9
                int 21h

                mov dx, offset NewLine
                mov ah, 9
                int 21h

            jmp OpenMenu

            ; the fallowing lines are the services

;==============================================================================================
        Exit:

        mov dx,offset S5
        mov ah,9
        int 21h

        mov ah,00h
        int 21h

;//////////////////////////////////////////////////////////////
                        jmp OpenMenu

;//////////////////////////////////////////////////////////////
```

```
Service1:

        word_have_deleted:

        again:
        mov dx, offset S1_length
        mov ah, 9
        int 21h

        mov ah, 1
        int 21h       ; After the user enter a number will check
                      ; the value that he entered.

                      ; start of condition
        mov bl, 3Ah   ; 3A = ':',after number 9 ('9' = 39h)
        cmp al, bl
        jae check
                      ; At this condition will check if value
                      ; if al is between 1 and 9 other than
                      ; that there will be an error message.

        mov bl, 2fh   ; 2F = '/' ,Before #0 ('0' = 30h)
        cmp al, bl
        jbe check     ; end of condition

        sub al, 30h
        mov ah, 0
        mov S1_number, ax

        mov dx, offset NewLine
        mov ah, 9
        int 21h

            jmp OpenMenu


        check:
        mov dx, offset NewLine
        mov ah, 9
        int 21h

        mov dx, offset NewLine
        mov ah, 9
        int 21h

        jmp again

;//////////////////////////////////////////////////////////////
                        jmp OpenMenu

;//////////////////////////////////////////////////////////////
```

```
        Service2:

        cmp [bx], 00         ; if not will ask the user to go to service--1--

        mov al , 9           ; 9 is the maximum number of word will take
        mov bx , S1_number   ; contain number of word
        mul bl
        mov S1_number, ax    ; value is 27

        mov bx, 0            ; bx will work as a counter
                             ; to exit when the user enter
                             ; the desired number of word
        lea si, word

        number_is_entered:

        mov dx, offset S2_word
        mov ah, 9
        int 21h

        fill_in:

              mov ah, 1
              int 21h

              cmp al, 0dh    ; if the user entered "enter" will go to next name
              je outer

              cmp al , 61h
              jb a_number_entered

              cmp al , 7ah
              ja a_number_entered


              mov [si], al   ; move the char to array word
              inc si

              loop fill_in

        outer:

        mov dx, offset NewLine
        mov ah, 9
        int 21h

        cmp si, S1_number
        jae near_OpenMenu    ; if SI is above or equal to number of word should finish the
enter processes
```

```
near_OpenMenu:
            mov S1_number, ax
            jmp OpenMenu
a_number_entered:

                 mov dx,offset NewLine
                 mov ah,9
                 int 21h

                 mov dx,offset S2_error
                 mov ah,9
                 int 21h


                 jmp number_is_entered
                 jmp near_OpenMenu

;//////////////////////////////////////////////////////////////
                        jmp OpenMenu
;//////////////////////////////////////////////////////////////

        Service3:

        begin:  mov cx,500
                mov ax,0b800h
                mov es,ax
                mov di,0
                mov al,word
                repne scasb       ;scan string byte by byte and repeat white
                                  ;not equal
                jz yes

                mov ah,9
                mov dx,offset S3_notfound
                mov ah,9
                int 21h

                mov dx,offset NewLine
                mov ah,9
                int 21h

            jmp OpenMenu

        yes:    mov ah,9
                mov dx,offset S3_found
                mov ah,9
                int 21h

                mov dx,offset NewLine
                mov ah,9
                int 21h
```

```
;///////////////////////////////////////////////////////////////
                        jmp OpenMenu
;///////////////////////////////////////////////////////////////

        Service4:

                lea si,S4_number
                mov cl,S4_counter
                mov ch,0

            mov cx,500
            mov ax,0b800h
            mov es,ax
            mov di,0

        find:
                mov al,[si]
                cmp word,al
                jne skip
                inc S4_count
        skip:
                inc si
                loop find

                cmp S4_count,0
                je notfound

                mov dl,S4_count
                add dl,30h
                mov ah,2
                int 21h

                mov ah,9
                lea dx,S4_found
                int 21h

            jmp OpenMenu

        notfound:

                mov ah,9
                lea dx,S4_notfound
                int 21h
;///////////////////////////////////////////////////////////////
                        jmp OpenMenu

;///////////////////////////////////////////////////////////////
    Main_prog endp
ends
end main_prog
```