KHULNA UNIVERSITY OF ENGINEERING AND TECHNOLOGY, KUET

SESSIONAL REPORT

Course No: CSE 2204

Department of: Computer Science and Engineering

Experiment No: **12**

Name of the Experiment: To Write an Assembly Language Program that takes a password input and stores 0001h in AX if the password is correct else stores 0000h in AX if the password is incorrect.

Remarks

Date of Performance: 24.05.21

Date of Submission: 05.06.21

Name: Rifat Arefin

Roll: 18070117

Year: 2nd

Semester: 2nd

No. of experiment: 12

Name of experiment: To develop an assembly program that inputs a password and stores 0001h in AX if the password is correct and 0000h in AX if it is incorrect.

Objectives:

1. To practice on string type data in assembly language.
2. To learn to compare string bytes
3. To learn about direction flags of 8086 microprocesson.
4. To execute chain instructions.

Introduction:

A string is a data-type used in program such as integer and floating point unit, but

is used to represent a group of characters. String is an example of a byte array and each character is presented as an ASCII code value (0 to 255).

Direction flag is a flag of 8086 that controls left to right or right to left direction of string processing. Generally the flag is used to determine the direction i.e forward or backword in which several byte of data will be copied from one place ~~to~~ in memory to another.

CLD is an instruction that clears direction flag. If it executes then SI and DI will automatically incremented by chain instructions such as cmpsb, lodsb, movsb etc.

STD is an instruction that set direction flag. STD executes to decrement SI and DI automatically by chain instructions i.g cmpsb, movsb.

CMPSB instruction compares bytes ES:[DI] to DS:[SI]

Algorithm:

DS : [SI] - ES : [DI]

Set flags according to result,

If DF = 0 then,

SI = SI + 1

DI = DI + 1

Else,

DI = DI - 1

SI = SI - 1

Apparatus required: emu 8086, laptop.

~~org 10~~

Methodology:

Code:

```
org 100h
.data
pass db 'kuetcse'      ; password to be matched
len equ ($-pass)       ; length of password.
str db len dup(0)      ; password to be given

.code
mov cx, len            ; initializing cx with len
lea si, pass           ; copy the address of pass to SI
lea di, str            ; copy the address of str to DI
cld                    ; clear direction flag
repe cmpsb             ; repeat if string bytes are
                       ; equal
jne incorrect          ; if input password doesn't matched
                       ; go to label 'incorrect'
```

```
je correct ; if the string bytes are equal then
           ; go to the label "correct"

incorrect:
        mov ax,0000h ; stores 0000h in ax
        ret

correct:
        mov ax, ~~0000~~ 0001h ; stores 0001h in ax
        ret
```

Result and Discussion:

In this program, we tried to check password from a given password. Actually in this program, we tried to check two strings that they are equal or not. If they were equal then we can say password matched. In this program we used many string instructions which are so much necessary for practicing programming in

assembly language. In this program we used various inputs and we came to a conclusion that our program performed well

Conclusion:

This experiment was very necessary to learn assembly language more perfectly. We learnt and implement various string ~~of~~ instruc-tions in this program. As we ensured that for every input our program performed well, so we can make conclusion that our implementation of these instruction worked successfully

References:

1. Microprocessor and Interfacing - by D. V. Hall
2. emu 8086/documentation/index.html