Northern University

of Business and Technology Khulna

Lab Report – 02

Course Code: CSE 3204

Course Title: Microprocessor and Assembly Language Programming Lab

Submitted By

Name	: Hemendra Nath Mondal
Section	: 6B
ID	: 11220320873
Dept	: Computer Science & Engineering

Submitted To

Name	: Md. Abdul Halim Khan
Designation	: Lecturer
Dept	: Computer Science & Engineering

Lab Report – 02

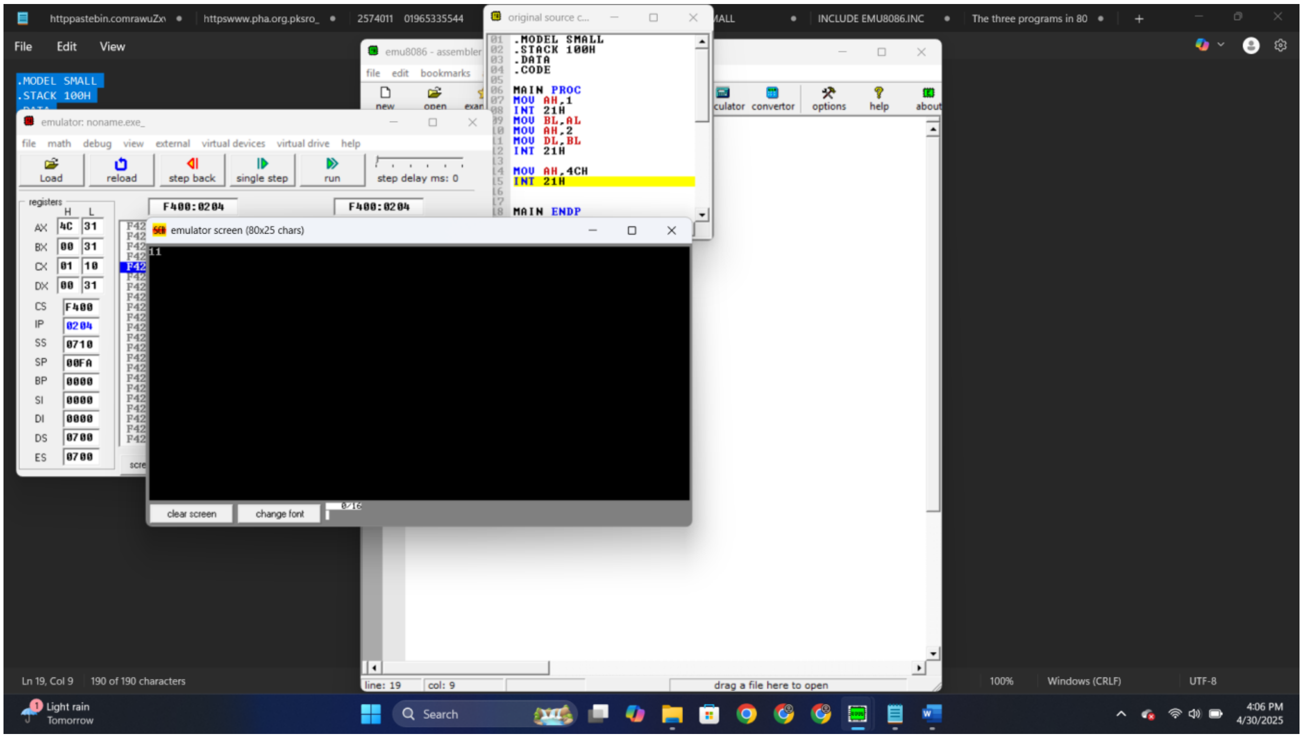
Description:

The three programs in 8086 Assembly demonstrate how to read a character from the user and display it back. The first program simply echoes the input character. The second program adds formatting by printing the character on a new line using carriage return and line feed. The third program improves readability by using the PRINTN macro from the EMU8086.INC file to automatically insert a newline. While the core functionality remains the same, each version introduces better output formatting and cleaner code structure, showcasing the use of MS-DOS interrupts and macros in assembly language. The three 8086 Assembly programs are designed to demonstrate the process of reading a character from the keyboard and displaying it back on the screen using MS-DOS interrupt services. The first program performs a basic task of reading a single character and immediately echoing it back without any additional formatting. The second program builds on this by introducing formatting to the output: after reading the character, it inserts a newline by using the carriage return (0DH) and line feed (0AH) functions before displaying the character. This ensures that the output appears on a new line, improving the visual clarity of the result. The third program takes advantage of the EMU8086.INC file and utilizes the PRINTN macro, which simplifies the code by automatically inserting a newline. This version not only provides the same functionality as the previous ones but also demonstrates how using macros can make the code more efficient and readable. Overall, these programs illustrate the basic concepts of character input and output in assembly language, while progressively enhancing formatting and code structure for better user experience and maintainability.

Code:

```
.MODEL SMALL
.STACK 100H
.DATA
.CODE

MAIN PROC
   MOV AH,1
   INT 21H
   MOV BL,AL
   MOV AH,2
   MOV DL,BL
   INT 21H

   MOV AH,4CH
   INT 21H


   MAIN ENDP
END MAIN
```

Output:



```
.MODEL SMALL
.STACK 100H
.DATA
.CODE

MAIN PROC
MOV AH,1
INT 21H
MOV BL,AL
MOV AH,2
MOV DL,BL
INT 21H

MOV AH,4CH
INT 21H


MAIN ENDP
```

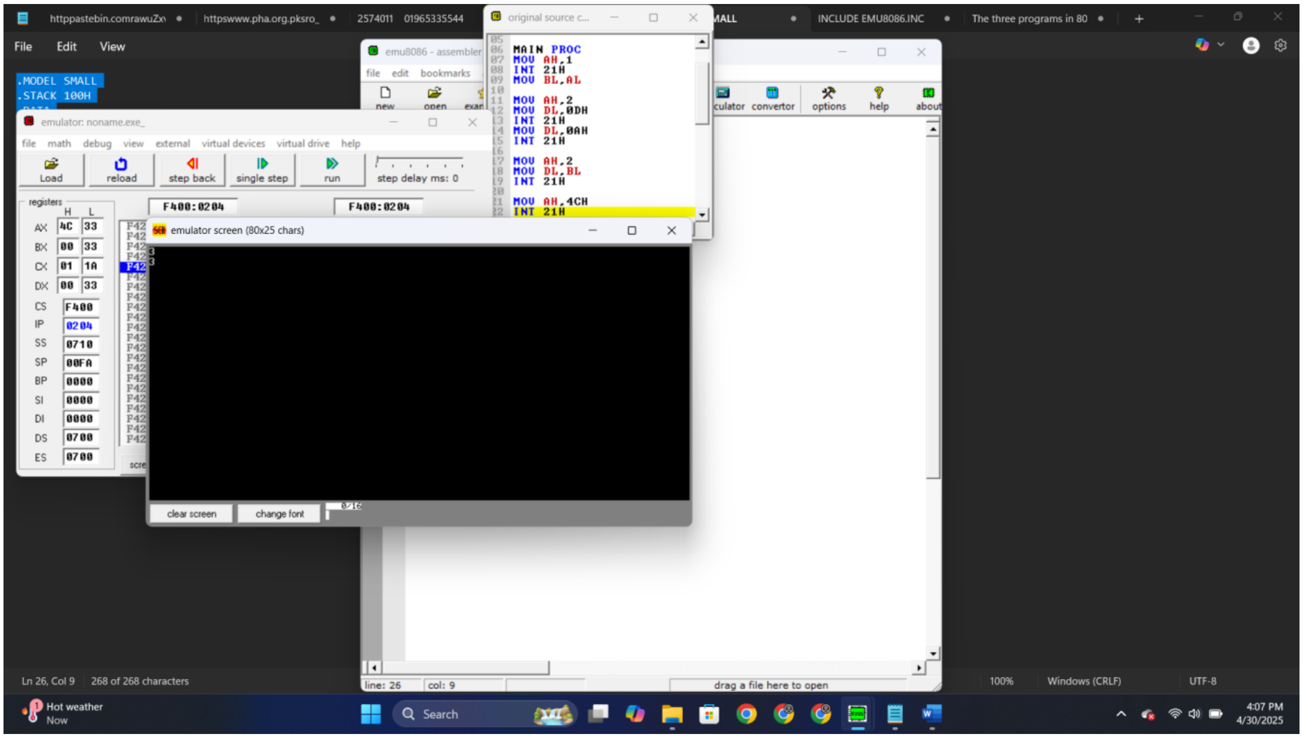
```
11
```

```
registers
      H   L
AX   4C  31
BX   00  31
CX   01  10
DX   00  31
CS   F400
IP   0204
SS   0710
SP   00FA
BP   0000
SI   0000
DI   0000
DS   0700
ES   0700
```

Code:

```
.MODEL SMALL
.STACK 100H
.DATA
.CODE

MAIN PROC
    MOV AH,1
    INT 21H
    MOV BL,AL

    MOV AH,2
    MOV DL,0DH
    INT 21H
    MOV DL,0AH
    INT 21H

    MOV AH,2
    MOV DL,BL
    INT 21H
    MOV AH,4CH
    INT 21H


    MAIN ENDP
END MAIN
```

Output:



```
.MODEL SMALL
.STACK 100H
```

```
MAIN PROC
MOV AH,1
INT 21H
MOV BL,AL

MOV AH,2
MOV DL,0DH
INT 21H
MOV DL,0AH
INT 21H

MOV AH,2
MOV DL,BL
INT 21H

MOV AH,4CH
INT 21H
```

emulator screen (80x25 chars)

```
3
3
```

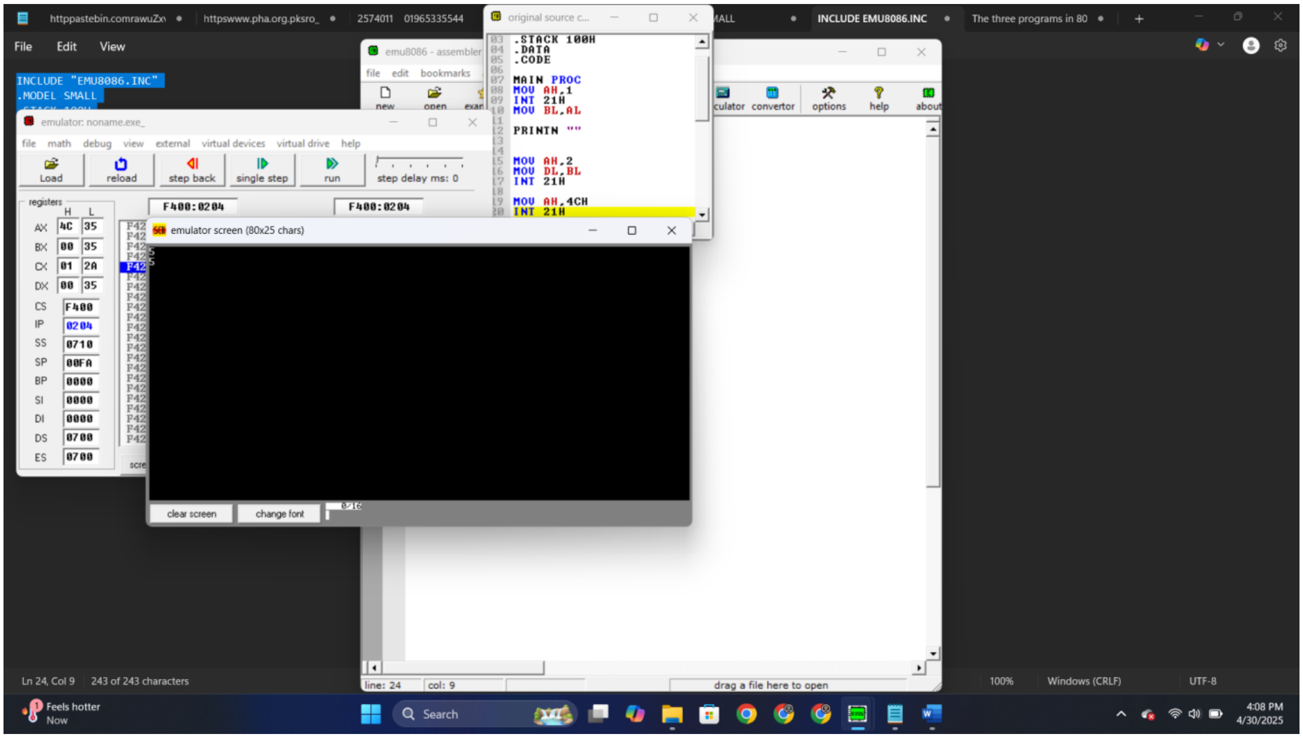
registers	H	L
AX	4C	33
BX	00	33
CX	01	1A
DX	00	33
CS	F400	
IP	0204	
SS	0710	
SP	00FA	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

F400:0204

Code:

```
INCLUDE "EMU8086.INC"
.MODEL SMALL
.STACK 100H
.DATA
.CODE

MAIN PROC
    MOV AH,1
    INT 21H
    MOV BL,AL

    PRINTN ""


    MOV AH,2
    MOV DL,BL
    INT 21H

    MOV AH,4CH
    INT 21H


    MAIN ENDP
END MAIN
```

Output:



```
INCLUDE "EMU8086.INC"
.MODEL SMALL
```

```
.STACK 100H
.DATA
.CODE

MAIN PROC
MOV AH,1
INT 21H
MOV BL,AL

PRINTN ""


MOV AH,2
MOV DL,BL
INT 21H

MOV AH,4CH
INT 21H
```

registers	H	L
AX	4C	35
BX	00	35
CX	01	2A
DX	00	35
CS	F400	
IP	0204	
SS	0710	
SP	00FA	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

```
5
5
```