

Assignment #03 – COAL – AI -3-1

Saghir Ali – 67005

Simple Calculator in EMU 8086:

This is a simple and efficient calculator program developed by Saghir Ali using x86 assembly language.

It provides basic arithmetic operations in a clean, menu-driven interface that is easy to use and understand.

Working:

The calculator starts by displaying Saghir Ali as name followed by a menu of available operations.

Users select their desired operation by entering corresponding numbers (1-4). The program then prompts for two single-digit inputs, performs the selected arithmetic operation, and instantly displays the calculated result.

The operations are

- 1. Addition**
- 2. Subtraction**
- 3. Multiplication**
- 4. Division**

CODE:

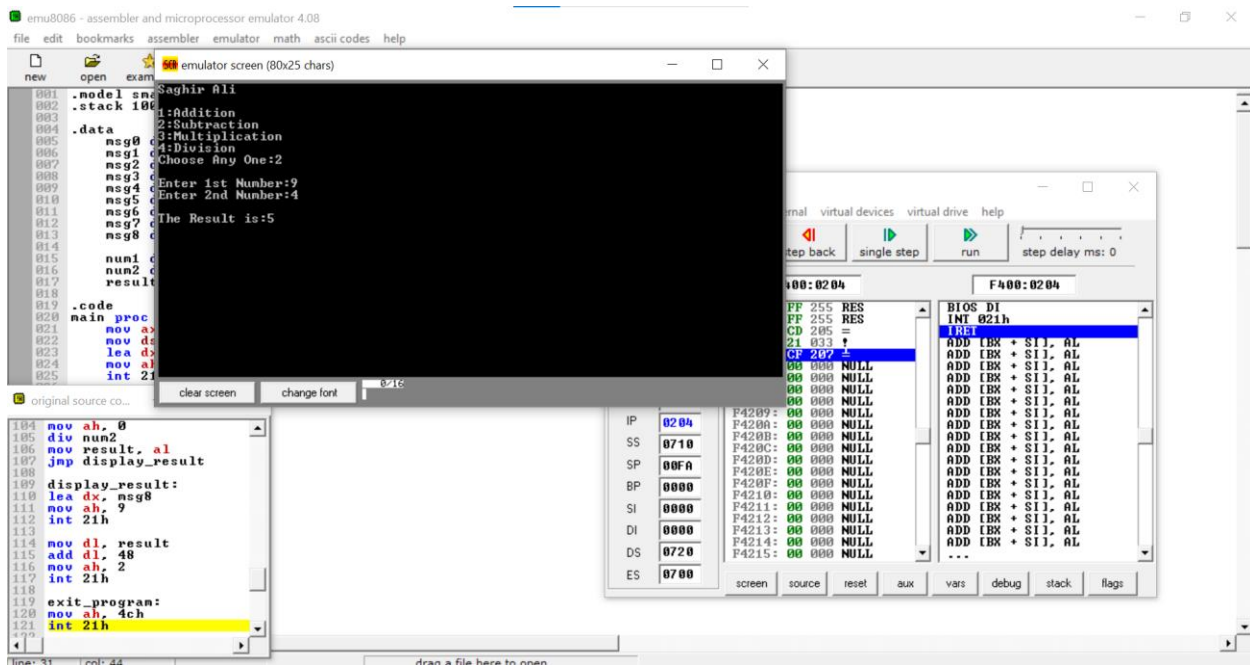
```
emu8086 - assembler and microprocessor emulator 4.08
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about

0001 .model small
0002 .stack 100h
0003
0004 .data
0005     msg0 db 'Saghir Ali$'
0006     msg1 db 10,13,10,13,'1:Addition$'
0007     msg2 db 10,13,'2:Subtraction$'
0008     msg3 db 10,13,'3:Multiplication$'
0009     msg4 db 10,13,'4:Division$'
0010     msg5 db 10,13,'Choose Any One:$'
0011     msg6 db 10,13,10,13,'Enter 1st Number:$'
0012     msg7 db 10,13,'Enter 2nd Number:$'
0013     msg8 db 10,13,10,13,'The Result is:$'
0014
0015     num1 db ?
0016     num2 db ?
0017     result db ?
0018
0019 .code
0020 main proc
0021     mov ax, @data
0022     lea dx, ax
0023     lea dx, msg0
0024     mov ah, 9
0025     int 21h
0026
0027     lea dx, msg1
0028     mov ah, 9
0029     int 21h
0030
0031     lea dx, msg2
0032     mov ah, 9
0033     int 21h
0034
0035     lea dx, msg3
0036     mov ah, 9
0037     int 21h
0038
0039     lea dx, msg4
0040     mov ah, 9
0041     int 21h
0042
0043     lea dx, msg5
0044     mov ah, 9
0045     int 21h
0046
0091     mov al, num1
0092     sub al, num2
0093     mov result, al
0094     jmp display_result
0095
0096 multiplication:
0097     mov al, num1
0098     mul num2
0099     mov result, al
0100     jmp display_result
0101
0102 division:
0103     mov al, num1
0104     mov ah, 0
0105     div num2
0106     mov result, al
0107     jmp display_result
0108
0109 display_result:
0110     lea dx, msg8
0111     mov ah, 9
0112     int 21h
0113
0114     mov dl, result
0115     add dl, 48
0116     mov ah, 2
0117     int 21h
0118
0119 exit_program:
0120     mov ah, 4ch
0121     int 21h
0122
0123 main endp
0124 end main

line: 69 col: 5 drag a file here to open
```

1. Addition

2. Subtraction:



[illegible][illegible]