

Actividad 5

Christian Geovany Muñoz Rodríguez

Ingeniería en computación

Código: 221350605

**Seminario de Solución de Problemas
Traductores de lenguajes I – D04 (Lunes y
Miércoles de 1 a 3)**

Maestro: José Juan Meza Espinosa

Universidad de Guadalajara

**Centro Universitario de Ciencias Exactas e
Ingenierías**

1 de marzo del 2023



Código:

```
org 100h
```

```
jmp inicio
```

```
;variables para el primer numero
```

```
unid1 db 0
```

```
de1 db 0
```

```
cen1 db 0
```

```
mill1 db 0
```

```
num1 dw 0
```

```
;variables para el segundo numero
```

```
unid2 db 0
```

```
de2 db 0
```

```
cen2 db 0
```

```
mill2 db 0
```

```
num2 dw 0
```

```
;variables para el resultado
```

```
unid3 db 0
```

```
de3 db 0
```

```
cen3 db 0
```

```
mill3 db 0
```

```
dcm db 0
```

```
num3 dw 0
```

```
;Mensajes de pantalla
```

```
msg db "Sumador y restador $"
```

```
msg2 db "Elija una opcion (s = suma / cualquier otro = resta): $"
```

```
msg3 db "Ingrese un numero: $"
```

```
msg4 db "El resultado es: $"
```

```
; Programa principal
```

```
inicio:
```

```
    LEA DX, msg
```

```
    MOV ah, 09h
```

```
    INT 21h
```

```
    MOV al, 10
```

```
    MOV ah,0eh
```

```
    INT 10h
```

```
    MOV al, 13
```

```
    MOV ah,0eh
```

```
    INT 10h
```

```
    LEA DX, msg2
```

```
    MOV ah, 09h
```

```
    INT 21h
```

MOV ah, 01h

INT 21h

CMP al, 's'

jz suma

jmp resta

ret

suma:

MOV al, 10

MOV ah, 0eh

INT 10h

MOV al, 13

MOV ah, 0eh

INT 10h

LEA DX, msg3

MOV ah, 09h

INT 21h

MOV ah, 01h

INT 21h

SUB al, 30h

MOV mill1, al ;Obtengo el digito de los millares del primer numero

MOV ah, 01h

INT 21h

SUB al, 30h

MOV cen1, al ;Obtengo el digito de las centenas del primer numero

MOV ah, 01h

INT 21h

SUB al, 30h

MOV de1, al ;Obtengo el digito de las decenas del primer numero

MOV ah, 01h

INT 21h

SUB al, 30h

MOV unid1, al ;Obtengo el digito de las unidades del primer numero

XOR ax, ax

MOV bx, 1000 ;Multiplico el digito de los millares por 1000

MOV al, mill1

MUL bx

MOV num1, ax ;Guardo el resultado en num1

XOR bx, bx ;Limpio el registro bx

MOV bl, 100 ;Multiplico el digito de las centenas por 100

MOV al, cen1

MUL bl

ADD num1, ax ;Sumo el resultado a num1

MOV bl, 10 ;Multiplico el digito de las decenas por 10

MOV al, de1

MUL bl

ADD num1, ax ;Sumo el resultado a num1

MOV bl, unid1 ;Sumo el digito de las unidades a num1

ADD num1, bx ;Sumo el resultado a num1

MOV al, 10

MOV ah, 0eh

INT 10h

MOV al, 13

MOV ah, 0eh

INT 10h

LEA DX, msg3

MOV ah, 09h

INT 21h

MOV ah, 01h ;Obtengo el digito de los millares del segundo numero

INT 21h

SUB al, 30h ;Convierto el caracter a numero

MOV mill2, al ;Guardo el digito en mill2

MOV ah, 01h ;Obtengo el digito de las centenas del segundo numero

INT 21h

SUB al, 30h ;Convierto el caracter a numero

MOV cen2, al ;Guardo el digito en cen2

MOV ah, 01h ;Obtengo el digito de las decenas del segundo numero

INT 21h

SUB al, 30h ;Convierto el caracter a numero

MOV de2, al ;Guardo el digito en de2

MOV ah, 01h ;Obtengo el digito de las unidades del segundo numero

INT 21h

SUB al, 30h

MOV unid2, al ;Guardo el digito en unid2

XOR ax, ax

MOV bx, 1000 ;Multiplico el digito de los millares por 1000

MOV al, mill2 ;Obtengo el digito de los millares del segundo numero

MUL bx

MOV num2, ax ;Guardo el resultado en num2

XOR bx, bx ;Limpio el registro bx

MOV bl, 100 ;Multiplico el digito de las centenas por 100

MOV al, cen2 ;Obtengo el digito de las centenas del segundo numero

MUL bl ;Multiplico el digito de las centenas por 100

ADD num2, ax ;Sumo el resultado a num2

MOV bl, 10 ;Multiplico el digito de las decenas por 10

MOV al, de2 ;Obtengo el digito de las decenas del segundo numero

MUL bl ;Multiplico el digito de las decenas por 10

ADD num2, ax ;Sumo el resultado a num2

MOV bl, unid2 ;Obtengo el digito de las unidades del segundo numero

ADD num2, bx ;Sumo el resultado a num2

MOV al, unid1 ;Sumo el digito de las unidades del primer numero

ADD al, unid2 ;Sumo el digito de las unidades del segundo numero

aaa

MOV unid3, al ;Guardo el resultado en unid3

MOV al, de1 ;Sumo el digito de las decenas del primer numero

ADC al, de2 ;Sumo el digito de las decenas del segundo numero

aaa

MOV de3, al ;Guardo el resultado en de3


```
MOV al, cen1 ;Sumo el digito de las centenas del primer numero
ADC al, cen2 ;Sumo el digito de las centenas del segundo numero
aaa
MOV cen3, al ;Guardo el resultado en cen3
```

```
MOV al, mill1 ;Sumo el digito de los millares del primer numero
ADC al, mill2 ;Sumo el digito de los millares del segundo numero
aaa
MOV mill3, al ;Guardo el resultado en mill3
```

```
MOV al, 0
ADC al, 0
aaa
MOV dcm, al ;Guardo el resultado en dcm
```

```
MOV al, 10
MOV ah, 0eh
INT 10h
```

```
;imprimo el resultado
MOV al, 13
MOV ah, 0eh
INT 10h
LEA DX, msg4
```

MOV ah, 09h

INT 21h

ADD unid3, 30h ;Convierto el numero a caracter

ADD de3, 30h ;Convierto el numero a caracter

ADD cen3, 30h ;Convierto el numero a caracter

ADD mill3, 30h ;Convierto el numero a caracter

ADD dcm, 30h ;Convierto el numero a caracter

MOV ah, 02h

MOV dl, dcm ;Imprimo el digito de las decenas de millar

INT 21h

MOV dl, mill3 ;Imprimo el digito de los millares

INT 21h

MOV dl, cen3 ;Imprimo el digito de las centenas

INT 21h

MOV dl, de3 ;Imprimo el digito de las decenas

INT 21h

MOV dl, unid3 ;Imprimo el digito de las unidades

INT 21h

ret

resta:

MOV al, 10

MOV ah, 0eh

INT 10h

MOV al, 13

MOV ah, 0eh

INT 10h

LEA DX, msg3

MOV ah, 09h

INT 21h

MOV ah, 01h

INT 21h

SUB al, 30h

MOV mill1, al ;Obtengo el digito de los millares del primer numero

MOV ah, 01h

INT 21h

SUB al, 30h

MOV cen1, al ;Obtengo el digito de las centenas del primer numero

MOV ah, 01h

INT 21h

SUB al, 30h

MOV de1, al ;Obtengo el digito de las decenas del primer numero

MOV ah, 01h

INT 21h

SUB al, 30h

MOV unid1, al ;Obtengo el digito de las unidades del primer numero

XOR ax, ax

MOV bx, 1000 ;Multiplico el digito de los millares por 1000

MOV al, mill1

MUL bx

MOV num1, ax ;Guardo el resultado en num1

XOR bx, bx ;Limpio el registro bx

MOV bl, 100 ;Multiplico el digito de las centenas por 100

MOV al, cen1

MUL bl

ADD num1, ax ;Sumo el resultado a num1

MOV bl, 10 ;Multiplico el digito de las decenas por 10

MOV al, de1

MUL bl

ADD num1, ax ;Sumo el resultado a num1

MOV bl, unid1 ;Sumo el digito de las unidades a num1

ADD num1, bx ;Sumo el digito de las unidades a num1

MOV al, 10

MOV ah, 0eh

INT 10h

MOV al, 13

MOV ah, 0eh

INT 10h

LEA DX, msg3

MOV ah, 09h

INT 21h

MOV ah, 01h

INT 21h

SUB al, 30h

MOV mill2, al ;Obtengo el digito de los millares del segundo numero

MOV ah, 01h

INT 21h

SUB al, 30h

MOV cen2, al ;Obtengo el digito de las centenas del segundo numero

MOV ah, 01h

INT 21h

SUB al, 30h

MOV de2, al ;Obtengo el digito de las decenas del segundo numero

MOV ah, 01h

INT 21h

SUB al, 30h

MOV unid2, al ;Obtengo el digito de las unidades del segundo numero

XOR ax, ax

MOV bx, 1000 ;Multiplico el digito de los millares por 1000

MOV al, mill2

MUL bx

MOV num2, ax ;Guardo el resultado en num2

XOR bx, bx ;Limpio el registro bx

MOV bl, 100 ;Multiplico el digito de las centenas por 100

MOV al, cen2

MUL bl

ADD num2, ax ;Sumo el resultado a num2

MOV bl, 10 ;Multiplico el digito de las decenas por 10

MOV al, de2

MUL bl

ADD num2, ax ;Sumo el resultado a num2

MOV bl, unid2 ;Sumo el digito de las unidades a num2

ADD num2, bx

MOV al, unid1 ;Resto el digito de las unidades del primer numero

SBB al, unid2

MOV unid3, al

MOV al, de1 ;Resto el digito de las decenas del primer numero

SBB al, de2

MOV de3, al

MOV al, cen1 ;Resto el digito de las centenas del primer numero

SBB al, cen2

MOV cen3, al

MOV al, mill1 ;Resto el digito de los millares del primer numero

SBB al, mill2

MOV mill3, al

MOV al, 0 ;Resto el digito de las decenas de millar del primer numero

SBB al, 0

MOV dcm, al ;Guardo el resultado en dcm

;Imprimo el resultado

MOV al, 10

MOV ah, 0eh

INT 10h

MOV al, 13

MOV ah, 0eh

INT 10h

LEA DX, msg4

MOV ah, 09h

INT 21h

;Convierto los numeros a caracter

ADD unid3, 30h

ADD de3, 30h

ADD cen3, 30h

ADD mill3, 30h

ADD dcm, 30h

MOV ah, 02h

;Imprimo el resultado

MOV dl, dcm

INT 21h

MOV dl, mill3

INT 21h

MOV dl, cen3


```
INT 21h

MOV dl, de3

INT 21h

MOV dl, unid3

INT 21h


ret
```

```
exit:

ret
```

Desarrollo

La primera tarea del programa es mostrar en pantalla un mensaje de bienvenida, para lo cual se usa la interrupción INT 21h con el valor de función 09h

Después de esto, se muestra en pantalla un mensaje solicitando al usuario que elija entre sumar o restar. Para ello, se utiliza nuevamente la interrupción INT 21h con el valor de función 09h. A continuación, se utiliza la interrupción INT 21h con la función 01h para leer el carácter ingresado por el usuario. Si el carácter es una "s", se salta a la etiqueta "suma", de lo contrario, se salta a la etiqueta "resta".

```
original source code
029 msg3 db "Ingrese un numero: $"
030 msg4 db "El resultado es: $"
031
032 ; Programa principal
033 inicio:
034 LEA DX, msg
035 MOV ah, 09h
036 INT 21h
037 MOV al, 10
038 MOV ah, 0eh
039 INT 10h
040
041 MOV al, 13
042 MOV ah, 0eh
043 INT 10h
044
045 LEA DX, msg2
046 MOV ah, 09h
047 INT 21h
048
049 MOV ah, 01h
050 INT 21h
```

```
emulator screen (80x25 chars)
Sumador y restador
Elija una opcion (s = suma / cualquier otro = resta): s
Ingrese un numero: _
```

En la etiqueta "suma" se capturan los números ingresados por el usuario y se almacenan en las variables correspondientes. Luego, se realiza la suma de los números y se muestra el resultado en pantalla.

```
original source code
065 suma:
066 MOV al, 10
067 MOV ah, 0eh
068 INT 10h
069
070 MOV al, 13
071 MOV ah, 0eh
072 INT 10h
073 LEA DX, msg3
074 MOV sh, 09h
075 INT 21h
076
077 MOV ah, 01h
078 INT 21h
079
080 SUB al, 30h
081 MOV mill1, al ;Obtengo el digito de los millares del primer numero
082 MOV ah, 01h
083 INT 21h
084 SUB al, 30h
085 MOV cen1, al ;Obtengo el digito de las centenas del primer numero
086 MOV ah, 01h
087 INT 21h
088 SUB al, 30h
089 MOV del, al ;Obtengo el digito de las decenas del primer numero
090 MOV ah, 01h
091 INT 21h
092 SUB al, 30h
093 MOV unid1, al ;Obtengo el digito de las unidades del primer numero
094
095 XOR ax, ax
096
097 MOV bx, 1000 ;Multiplico el digito de los millares por 1000
098 MOV al, mill1
099 MUL bx
100 MOV num1, ax ;Guardo el resultado en num1
101 XOR bx, bx ;Limpio el registro bx
102
103 MOV bl, 100 ;Multiplico el digito de las centenas por 100
104 MOV al, cen1
105 MUL bl
106 ADD num1, ax ;Sumo el resultado a num1
107
108 MOV bl, 10 ;Multiplico el digito de las decenas por 10
109 MOV al, del
110 MUL bl
111 ADD num1, ax ;Sumo el resultado a num1
112
113 MOV bl, unid1 ;Sumo el digito de las unidades a num1
114 ADD num1, bx ;Sumo el resultado a num1
115 MOV al, 10
116 MOV ah, 0eh
117 INT 10h
118
119 MOV al, 13
120 MOV ah, 0eh
121 INT 10h
122 LEA DX, msg3
```

```
emulator screen (80x25 chars)
Sumador y restador
Elija una opcion (s = suma / cualquier otro = resta): s
Ingrese un numero: 8778
Ingrese un numero: 2334
El resultado es: 11112

clear screen  change font  0/16
```

En la etiqueta "resta" se realiza una tarea similar, pero en lugar de sumar, se resta. Finalmente, después de mostrar el resultado en pantalla, el programa termina.

```
ret
restai: 10
MOV al, 10
MOV ah, 0eh
INT 10h
MOV al, 13
MOV ah, 0eh
INT 10h
LEA dx, msg3
MOV sh, 09h
INT 21h
MOV ah, 01h
INT 21h
SUB al, 30h
MOV milli, al ;Obtengo el digito de los millares del primer numero
MOV ah, 01h
INT 21h
SUB al, 30h
MOV cent, al ;Obtengo el digito de las centenas del primer numero
MOV ah, 01h
INT 21h
SUB al, 30h
MOV dec, al ;Obtengo el digito de las decenas del primer numero
MOV ah, 01h
INT 21h
SUB al, 30h
MOV unidi, al ;Obtengo el digito de las unidades del primer numero ;Obtengo el digito de las unidades del primer numero
XOR ax, ax
MOV bx, 1000 ;Multiplico el digito de los millares por 1000
MOV al, milli
MUL bx
MOV num1, ax ;Guardo el resultado en num1
XOR bx, bx ;Limpio el registro bx
MOV bl, 100 ;Multiplico el digito de las centenas por 100
MOV al, cent
MUL bl
ADD num1, ax ;Sumo el resultado a num1
MOV bl, 10 ;Multiplico el digito de las decenas por 10
MOV al, dec
MUL bl
ADD num1, ax ;Sumo el resultado a num1
MOV bl, unidi ;Sumo el digito de las unidades a num1
ADD num1, bx ;Sumo el digito de las unidades a num1
MOV al, 10
INT 10h
MOV al, 13
MOV ah, 0eh
INT 10h
```

```
emulator screen (80x25 chars)
Sumador y restador
Elija una opcion (s = suma / cualquier otro = resta): r
Ingrese un numero: 4567
Ingrese un numero: 3200
El resultado es: 01367
clear screen change font 0/16
```

En resumen, este programa es un sumador/restador de números de cuatro dígitos. El usuario elige entre sumar o restar, ingresa los números a operar, y el programa muestra el resultado en pantalla.

Conclusiones:

Realizar este programa me resultó una buena experiencia porque:

Me permitió entender cómo se manejan los datos y se realizan operaciones aritméticas más complejas a bajo nivel en un lenguaje de programación cercano al hardware.

Este tipo de programa es útil en muchos contextos prácticos, como en el ámbito de la electrónica y la robótica, donde se requiere el manejo preciso de datos numéricos y operaciones matemáticas complejas. También es útil para entender el funcionamiento de otros programas y sistemas que se basan en operaciones aritméticas, como los programas de contabilidad y finanzas, por ejemplo.

Bibliografía:

Brey, B. B. (2006). *Microprocesadores Intel : 8086/8088, 80186/80188, 80286, 80386 y*

80486, Pentium, procesador Pentium Pro, Pentium II, Pentium III y Pentium 4:

arquitectura, programación e interfaces. Pearson Educación.