

Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Навчально-науковий Фізико-технічний інститут

Архітектура комп'ютерних систем
Комп'ютерний практикум
Робота №2

Виконав:
студент групи ФІ-12
Завалій Олександр
Перевірив:
Козленко О.В.

Робота №2.

Основи побудови програми на асемблері в архітектурі IA-32

Мета:

Ознайомитися з створенням базової програми виключно на мові асемблер для платформи на архітектурі IA-32

Варіант №4

Зміст індивідуального завдання:

1. Визначити дані.
 $a(1) \rightarrow 8, a(2) \rightarrow 5, a(3) \rightarrow 3, c1 \rightarrow 20, c2 \rightarrow 6$
2. Занести в регістри такі величини.
 $AX \rightarrow a(1) + a(2) - a(3), BX \rightarrow a(1) \cdot a(2), CX \rightarrow c1 - c2, DX \rightarrow ((c1 \& a(2)), a(3))$
3. Організувати цикл, послідовно зменшуючи число у регістрі CX на 1. У циклі зменшувати число, що знаходиться у регістрі BX на величину, що знаходиться у регістрі AX , поки значення CX не стане дорівнювати 0.

Code

Код на m0vi Assembler.

```
section .bss
    val_AX resd 1
    val_BX resd 1
    val_CX resd 1
    val_DX resd 1

section .data
    a db 8, 5, 3
    c1 db 20
    c2 db 6
    fmt db "Result: %d", 10, 0

section .text
    extern printf
    global main

main:
    push ebp
    mov ebp, esp

    ; AX = a(1) + a(2) - a(3)
    mov al, [a]
    add al, [a+1]
    sub al, [a+2]
    movzx eax, al
    mov [val_AX], eax

    ; BX = a(1) * a(2)
    mov al, [a]
    mov bl, [a+1]
    movzx eax, al
    movzx ebx, bl
    imul eax, ebx
    mov [val_BX], eax

    ; CX = c1 - c2
    mov al, [c1]
    sub al, [c2]
    movzx eax, al
    mov [val_CX], eax

    ; DX = ((c1 & a(2)), a(3))
    mov al, [c1]
    and al, [a+1]
    add al, [a+2]
    movzx eax, al
    mov [val_DX], eax
```

```

; Loop
loop_start:
    mov eax, [val_CX]
    cmp eax, 0
    je loop_end

    ; CX = CX - 1
    sub dword [val_CX], 1

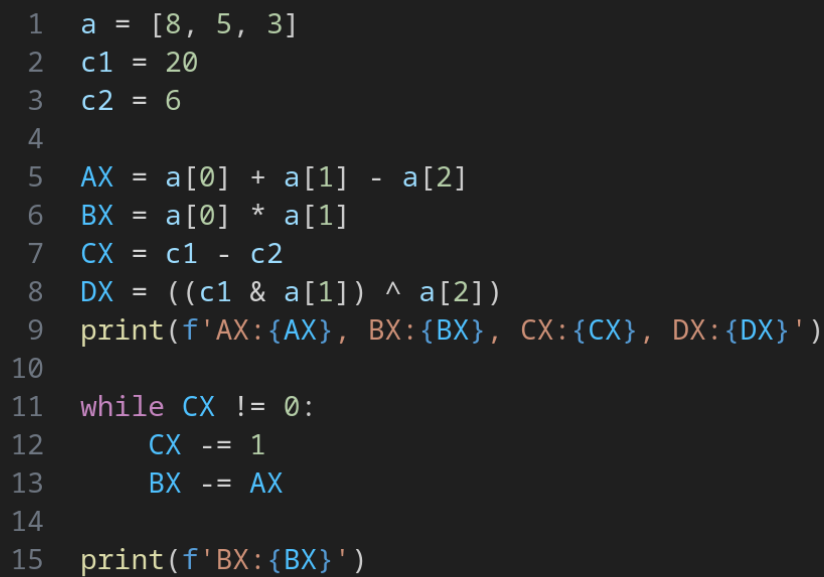
    ; BX = BX - AX
    mov eax, [val_BX]
    sub eax, [val_AX]
    mov [val_BX], eax
    jmp loop_start

loop_end:
    push dword [val_BX]
    push fmt
    call printf
    add esp, 8

leave
ret

```

Для перевірки коректності обрахунків я написав код на мові Python.



```

1  a = [8, 5, 3]
2  c1 = 20
3  c2 = 6
4
5  AX = a[0] + a[1] - a[2]
6  BX = a[0] * a[1]
7  CX = c1 - c2
8  DX = ((c1 & a[1]) ^ a[2])
9  print(f'AX:{AX}, BX:{BX}, CX:{CX}, DX:{DX}')
10
11 while CX != 0:
12     CX -= 1
13     BX -= AX
14
15 print(f'BX:{BX}')

```

Results

```

1 section .bss
2     val_AX resd 1
3     val_BX resd 1
4     val_CX resd 1
5     val_DX resd 1
6
7 section .data
8     a db 8, 5, 3
9     c1 db 20
10    c2 db 6
11    fmt db "Result: %d", 10, 0
12
13 section .text
14     extern printf
15     global main
16
17 main:
18     push ebp
19     mov ebp, esp
20
21     ; AX = a(1) + a(2) - a(3)
22     mov al, [a]
23     add al, [a+1]
24     sub al, [a+2]
25     movzx eax, al
26     mov [val_AX], eax
27
28     ; BX = a(1) * a(2)
29     mov al, [a]
30     mov bl, [a+1]
31     movzx eax, al
32     movzx ebx, bl
33     imul eax, ebx
34     mov [val_BX], eax
35
36     ; CX = c1 - c2
37     mov al, [c1]
38     sub al, [c2]
39     movzx eax, al
40     mov [val_CX], eax
41
42     ; DX = ((c1 & a(2)), a(3))
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71

```


[16:56:32] Build started...
[16:56:32] Built successfully.
[16:56:33] The program is executing...
[16:56:33] The program finished normally. Execution time: 0.011 s

```

31     movzx eax, al
32     movzx ebx, bl
33     imul eax, ebx
34     mov [val_BX], eax
35
36     ; CX = c1 - c2
37     mov al, [c1]
38     sub al, [c2]
39     movzx eax, al
40     mov [val_CX], eax
41
42     ; DX = ((c1 & a(2)), a(3))
43     mov al, [c1]
44     and al, [a+1]
45     add al, [a+2]
46     movzx eax, al
47     mov [val_DX], eax
48
49     ; Loop
50 loop_start:
51     mov eax, [val_CX]
52     cmp eax, 0
53     je loop_end
54
55     ; CX = CX - 1
56     sub dword [val_CX], 1
57
58     ; BX = BX - AX
59     mov eax, [val_BX]
60     sub eax, [val_AX]
61     mov [val_BX], eax
62     jmp loop_start
63
64 loop_end:
65     push dword [val_BX]
66     push int
67     call printf
68     add esp, 8
69
70     leave
71     ret

```

[16:56:32] Build started...
[16:56:32] Built successfully.
[16:56:33] The program is executing...
[16:56:33] The program finished normally. Execution time: 0.011 s



The screenshot shows a dark-themed interface with a horizontal menu at the top containing 'PROBLEMS' (with a '29' badge), 'OUTPUT' (highlighted with a blue underline), 'DEBUG CONSOLE', 'TERMINAL', 'PORTS', 'JUPYTER', and 'GITLENS'. Below the menu, the output text is as follows:

```
[Running] python -u "/home/alex/Projects/ACS/lab 2/Code/main.py"  
AX:10, BX:40, CX:14, DX:7  
BX:-100  
  
[Done] exited with code=0 in 0.029 seconds
```

Результати роботи програм співпадають.