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 №4

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

Київ-2024

Робота №4.
Основи побудови програми на асемблері в архітектурі x64 в операційній системі Windows з використанням WinAPI

Мета:

Ознайомитися з створенням базової програми мовою асемблер для операційної системи Windows 10 (або пізніше) архітектури x64 з використанням NASM.

Варіант №4

Зміст індивідуального завдання:

- Визначити дані:
  $a(1) \to 12$, $a(2) \to 6$, $a(3) \to 17$, $c1 \to 23$, $c2 \to 16$
- Занести в регістри такі величини:
  $RAX \to a(1) + a(3) - a(2)$, $RBX \to \frac{a(1)}{a(2)}$, $RCX \to c1 + c2$, $RDX \to a\,{}^{rc\to 4}_{1}$
- Організувати цикл, послідовно зменшуючи число у регістрі RCX на 5. У циклі збільшувати число, що знаходиться у регістрі RBX на величину, що знаходиться у регістрі RAX, та зменшувати на величину, що знаходиться у регістрі RCX, поки значення RCX не стане менше 5.
- Створити MessageBox та записати у нього відповідні значення регістру RCX після всіх операцій.

Code

```
global start
NULL equ 0
MB_OK equ 0
     extern MessageBoxA
     extern ExitProcess

section .data
    a1 equ 12
    a2 equ 6
    a3 equ 17
    c1 equ 23
    c2 equ 16
    text: db 'rcx: ',0
    title: db 'Result',0

section .bss
    numbuf resb 20

section .text

itoa:
    push rbp
    mov rbp,rsp
    sub rsp,8
    mov rax,rcx
    lea rdi,[numbuf+10]
    mov rcx,10
    mov qword [rbp-8],0

.divloop:
    xor rdx,rdx
    idiv rcx
    add rdx,0x30
    dec rdi
    mov byte [rdi],dl
    inc qword [rbp-8]
    cmp rax,0
    jnz .divloop
    mov rax,rdi
    leave
    ret
```

```
start:
     mov rax, a1
     add rax, a3
     sub rax, a2
     push rax
     mov rdx, 0
     mov rax, a1
     mov rbx, a2
     div rbx
     mov rbx, rax
     pop rax
     mov rcx, c1
     add rcx, c2
     mov rdx, a1
     rcr rdx, 4
loop:
     cmp rcx, 5
     jl loopend
     sub rcx, 5
     add rbx, rax
     sub rbx, rcx
     jmp loop

loopend:
     call itoa
     mov rcx, rbx
     mov rbx, [rax]
     mov [text+5], rbx
     sub rsp, 40
     mov r9, MB_OK
     mov r8, title
     mov rdx, text
     mov rcx, NULL
     call MessageBoxA

quit:
     xor rcx, rcx
     call ExitProcess
```

```
a = [12, 6, 17]
c1 = 23
c2 = 16
```
[4] ✓ 0.0s Python

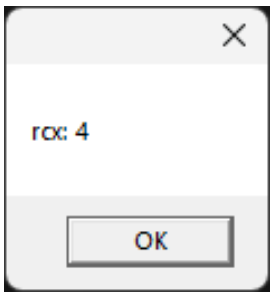
```
RAX = a[0] + a[2] - a[1]
RBX = a[0] / a[1]
RCX = c1 + c2
```
[3] ✓ 0.0s Python

```
while RCX >= 5:
    RCX -= 5
    RBX += RAX - RCX
```
[5] ✓ 0.0s Python

```
RCX
```
[6] ✓ 0.0s Python

```
4
```

Results



Результати роботи програм співпадають.