

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«Национальный исследовательский университет
Московский институт электронной техники»

**ЛАБОРАТОРНАЯ РАБОТА № 6
ПО АРХИТЕКТУРАМ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ
РАБОТА С УСЛОВНЫМИ КОМАНДАМИ**

ВАРИАНТ 6

Егоров Вадим, Кудреватых Павел, Саркисов Эрик
ПМ-31

Январь 2022 г.

Содержание

Задание Л6.№1..	1
Задание Л6.№2..	1
Задание Л6.№3..	2
Задание Л6.№4..	3
Задание Л6.№5..	5
Задание Л6.№6..	7

Цель работы: ознакомиться с командами условной пересылки, условной установки байта, а также условного и безусловного перехода: научиться использовать их для реализации ветвящихся и циклических алгоритмов.

Задание Л6.№1.

Вычислите для заданного целого знакового x :

Вариант 6: $f(x) = (x \neq 9)$

```
global main
extern printf

section .text
str: db "Result: %d", 10, 0
x dd 8

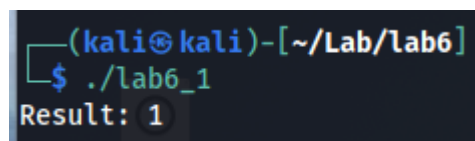
main:      sub rsp, 8

           xor rax, rax
           cmp dword [x], 9
           setne al

           mov rdi, str
           mov rsi, rax
           call printf

           add rsp, 8
           xor rax, rax
           ret
```

Вывод:



```
(kali㉿kali)-[~/Lab/lab6]
$ ./lab6_1
Result: 1
```

Задание Л6.№2.

Вычислите для заданного целого знакового x : Вариант 6:

$$y(x) = \begin{cases} 4, & x < 3 \\ 8x - 52, & \text{иначе} \end{cases}$$

```

global main
extern printf

section .text
str: db "Result: %d", 10, 0
x dq 0

asm2:      cmp rdi, 3
           jl var_1

           mov rax, rdi
           mov rbx, 8
           xor rdx, rdx
           mul rbx
           sub rax, 52
           jmp exit

var_1:      mov rax, 4

exit:      ret

main:      sub rsp, 8

           mov rdi, [x]
           call asm2

           mov rdi, str
           mov rsi, rax
           call printf

           add rsp, 8
           xor rax, rax
           ret

```

Вывод:



```

(kali㉿kali)-[~/Lab/lab6]
$ ./lab6_2
Result: 4

```

Задание Л6.№3.

Вычислите для заданного вещественного x

$$y(x) = \begin{cases} 4,6, & x < 3,3 \\ x + 6,9, & \text{иначе} \end{cases}$$

```

global main
extern printf

section .text
str: db "Result: %lf", 10, 0
x dq 3.2
v1 dq 4.6
v2 dq 6.9
v_cmp dq 3.3

```

```

asm3: push rbp
      mov rbp, rsp

      sub rsp, 8
      movq [rsp], xmm0

      fld qword [rsp]      ;x
      fcom qword [v_cmp]
      fstsw ax
      sahf
      jnb var_1

      fld qword [v2]      ;6.9
      faddp                ;+

      fstp qword [rsp]
      movq xmm0, [rsp]
      jmp exit

var_1: movq xmm0, [v1]    ;4.6

exit: mov rsp, rbp
      pop rbp
      ret

main: sub rsp, 8

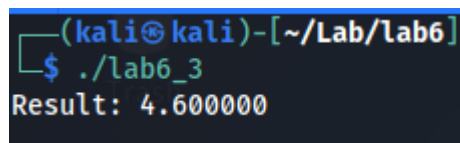
      movq xmm0, [x]
      call asm3

      mov rdi, str
      mov al, 1
      call printf

      add rsp, 8
      xor rax, rax
      ret

```

Вывод:



```

(kali㉿kali)-[~/Lab/lab6]
└─$ ./lab6_3
Result: 4.600000

```

Задание Л6.№4.

Задайте с клавиатуры N и x_0 и напечатайте первые N членов целочисленной последовательности:

$$x_{i+1} = \begin{cases} x_i/2, & x_i \% 2 = 0 \\ 3x_i + 1, & x_i \% 2 \neq 0 \end{cases}.$$

```

global main
extern scanf, printf

section .bss
x0          resq 1
N           resq 1

section .text

```

```

str_printf_1      db "Enter x0 (>=0): ", 0
str_printf_2      db "Enter N (>=0): ", 0
str_scanf         db "%d", 0
str_printf_3      db "%d", 0
str_printf_4      db 10, 0

```

```

scan:             mov rdi, str_printf_1
                  mov al, 0
                  call printf

                  mov rdi, str_scanf
                  mov rsi, x0
                  mov al, 0
                  call scanf

                  mov rdi, str_printf_2
                  mov al, 0
                  call printf

                  mov rdi, str_scanf
                  mov rsi, N
                  mov al, 0
                  call scanf

                  ret

```

```

asm4:             push rbx

                  mov rax, rdi
                  mov rbx, 2
                  xor rdx, rdx
                  div rbx
                  cmp rdx, 0      ;  $x_i \% 2 == 0$ 
                  je var_1        ;  $x_{i+1} = x_i / 2$ 

                  mov rax, rdi
                  mov rbx, 3
                  mul rbx
                  add rax, 1       ;  $x_{i+1} = 3 * x_i + 1$ 

```

```

var_1:            pop rbx
                  ret

```

```

main:             call scan

                  cmp qword [N], 0 ;  $N == 0$ 
                  je exit

                  mov rdi, str_printf_3
                  mov rsi, [x0]
                  mov al, 0
                  call printf      ; print x0

                  mov rbx, [x0]    ; x0
                  mov rcx, [N]     ; counter

                  dec rcx          ; counter--
                  jz exit          ; counter == 0

```

```

next:      push rcx

           mov rdi, rbx
           call asm4
           mov rbx, rax

           mov rdi, str_printf_3
           mov rsi, rbx
           mov al, 0
           call printf

           pop rcx

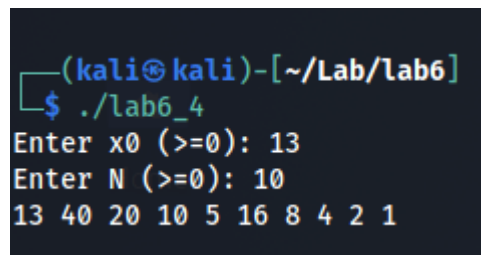
           dec rcx          ;counter--
           jnz next        ;counter!=0

exit:      mov rdi, str_printf_4
           mov al, 0
           call printf

           xor rax, rax
           ret

```

Вывод:



```

(kali@kali)-[~/Lab/lab6]
$ ./lab6_4
Enter x0 (>=0): 13
Enter N (>=0): 10
13 40 20 10 5 16 8 4 2 1

```

Задание Л6.№5.

Опишите вставку либо функцию, инициализирующую целочисленный массив заданной длины N первыми N элементами последовательности из задания Л6.№4.

```

#include <stdio.h>
extern void asm5(unsigned int*, unsigned int, unsigned int);

int main()
{
    unsigned int N = 0, x0 = 0;
    printf("Enter x0 and N: ");
    scanf("%d %d", &x0, &N);

    unsigned int array[N];
    asm5(array, x0, N);

    printf("Result: ");
    int i = 0;
    for(i; i < N; i++)
        printf("%d ", array[i]);
    printf("\n");

    return 0;
}

```

```

global asm5

asm4:      push rbx

            mov rax, rdi
            mov rbx, 2
            xor rdx, rdx
            div rbx
            cmp rdx, 0          ;x_i%2==0
            je var_1           ;x_(i+1) = x_i/2

            mov rax, rdi
            mov rbx, 3
            mul rbx
            add rax, 1          ;x_(i+1) = 3*x_i+1

var_1:      pop rbx
            ret

asm5:      push rdx             ;N
            mov rax, rsi        ;x0
            xor rcx, rcx        ;counter

            cmp rcx, [rsp]      ;N == 0
            je exit

            mov [rdi], rax      ;x0 -> array

            inc rcx             ;counter++
            cmp rcx, [rsp]
            jz exit             ;counter == N

next:      push rcx
            push rdi

            mov rdi, rax
            call asm4

            pop rdi
            pop rcx

            mov [rdi + rcx*4], rax

            inc rcx
            cmp rcx, [rsp]
            jne next           ;counter != N

exit:      pop rdx
            ret

```

Вывод:

```

(kali@kali)-[~/Lab/lab6]
$ ./lab6_5
Enter x0 and N: 13 10
Result: 13 40 20 10 5 16 8 4 2 1

```

Задание Л6.№6.

Реализуйте задание Л5.№2 лабораторной работы Л5 (расчёт суммы лейбницевского ряда) как функцию на ассемблере.

```
global main
extern printf

section .text
e          dq 0.000001
str:       db "Result: %lf", 10, 0

value:     push rbx

            sub rsp, 8
            mov [rsp], rdi

            fild qword [rsp]      ; i
            fld1                   ; 1
            faddp                   ; +

            fild qword [rsp]      ; i
            fild qword [rsp]      ; i
            fmulp                   ; *
            fld1                   ; 1
            fsubp                   ; -

            fdivp                   ; /

            mov rax, rdi
            xor rdx, rdx
            mov rbx, 2
            div rbx
            cmp rdx, 0
            je exit                ; xmm0

            fld st0
            fsub st1, st0
            fsubp                   ; -xmm0

exit:       fstp qword [rsp]
            movq xmm0, [rsp]
            add rsp, 8

            pop rbx
            ret

main:       xorps xmm1, xmm1 ; s
            xorps xmm2, xmm2 ; s_1
            xorps xmm3, xmm3 ; s_2
            xorps xmm4, xmm4 ; error

            sub rsp, 8
            mov rcx, 2          ; i = 2

next:       mov rdi, rcx
            call value          ; value(i)

            vmovapd xmm2, xmm1 ; s_1 = s
            vmovapd xmm3, xmm0 ; s_2 = value(i)
            vaddpd xmm1, xmm1, xmm3 ; s += s_2
```



```

vaddpd xmm4, xmm4, xmm3
vsubpd xmm4, xmm4, xmm1
vaddpd xmm4, xmm4, xmm2
vaddpd xmm4, xmm4, xmm2
vsubpd xmm4, xmm4, xmm1
vaddpd xmm4, xmm4, xmm1
vsubpd xmm4, xmm4, xmm2 ;error

inc rcx ;i++
mov rdi, rcx
call value ;value(i+1)

movq [rsp], xmm0
fld qword [rsp]

mov rax, rcx
xor rdx, rdx
mov rbx, 2
div rbx
cmp rdx, 0
je pos

fld st0
fsub st1, st0
fsubp

pos:      fld qword [e]
fcompp
fstsw ax
sahf
ja stop

jmp next

stop:     mov rdi, str
vaddpd xmm0, xmm1, xmm4 ;res = s + error
mov al, 1
call printf

add rsp, 8
xor rax, rax
ret

```

Вывод:

```

(kali@kali)-[~/Lab/lab6]
$ ./lab6_6
Result: 0.693147

```