

---

## Лабораторная №3.

---

### Модули и функции на Ассемблере.

Выполнили: Шигирданова Анна  
Ишков Григорий  
Егорова Юлия  
ПИН-31

**Задание 1.** Разработайте ассемблерную функцию, вычисляющую целое выражение от целого аргумента (в соответствии с вариантом), а также головную программу на языке C++, использующую разработанную функцию.

2	$y(x) = x \% 4 - x$
---	---------------------

```
#include <iostream>
#include <math.h>
#include <string>

using namespace std;

extern "C" int main1 (int x);

int main ()
{
    for(int i = 0; i < 30; i++)
    {
        cout << "x: " << i << "\tC++: " << (i%4-i) << "\tAsm: " << main1(i) << endl;
    }
    return 0;
}
```

```
.global _main1 /*точка входа в программу*/
_main1:
    mov %ecx, %eax /*ecx = eax*/
    shr $2, %eax /*сдвиг влево (eax = eax << 2)*/
    shl $2, %eax /*беззнаковый сдвиг вправо (eax=(unsigned)eax >> 2)*/
    imul $-1, %eax /*умножение (eax*=-1)*/
    ret /*возврат из программы - снимает со стека адрес возврата и помещает его в указатель команд*/
```

```
H:\avsLab3\asm1\bin\Debug\asm1.exe
x: 0 C++: 0 Asm: 0
x: 1 C++: 0 Asm: -2130563072
x: 2 C++: 0 Asm: -2130563072
x: 3 C++: 0 Asm: -2130563072
x: 4 C++: -4 Asm: -2130563072
x: 5 C++: -4 Asm: -2130563072
x: 6 C++: -4 Asm: -2130563072
x: 7 C++: -4 Asm: -2130563072
x: 8 C++: -8 Asm: -2130563072
x: 9 C++: -8 Asm: -2130563072
x: 10 C++: -8 Asm: -2130563072
x: 11 C++: -8 Asm: -2130563072
x: 12 C++: -12 Asm: -2130563072
x: 13 C++: -12 Asm: -2130563072
x: 14 C++: -12 Asm: -2130563072
x: 15 C++: -12 Asm: -2130563072
x: 16 C++: -16 Asm: -2130563072
x: 17 C++: -16 Asm: -2130563072
x: 18 C++: -16 Asm: -2130563072
x: 19 C++: -16 Asm: -2130563072
x: 20 C++: -20 Asm: -2130563072
x: 21 C++: -20 Asm: -2130563072
x: 22 C++: -20 Asm: -2130563072
x: 23 C++: -20 Asm: -2130563072
x: 24 C++: -24 Asm: -2130563072
x: 25 C++: -24 Asm: -2130563072
x: 26 C++: -24 Asm: -2130563072
x: 27 C++: -24 Asm: -2130563072
x: 28 C++: -28 Asm: -2130563072
x: 29 C++: -28 Asm: -2130563072

Process returned 0 (0x0)   execution time : 0.062 s
Press any key to continue.
```

**Задание 2.** Разработайте программу, целиком написанную на ассемблере, вычисляющую значение  $y(x)$  для  $x = 13$  и выводящую полученное значение на стандартный вывод с использованием библиотеки libc (в частности, функции printf).

```
.data
    msg: .string "%d mod %d - 4 = %d\n"
    x: .int 13

.global _main
_main:
    mov x, %eax
    shr $2, %eax
    shl $2, %eax
    imul $-1, %eax

    push %eax
    push x
    push x
    push $msg
    call _printf

    addl $4*4, %esp
    xor %eax, %eax
ret
```

```
H:\avsLab3\asm2\bin\Debug\asm2.exe
13 mod 13 - 4 = -12
Process returned 0 (0x0) execution time : 0.031 s
Press any key to continue.
```

**Задание 3.** Опишите функцию на произвольном языке высокого уровня (включая C/C++) и вызовите её из ассемблерной функции.

$(N - 1) \% 3 + 1$	Вариант
1	Вывод двух параметров на экран с пояснениями
2	Ввод результата с клавиатуры
3	Случайный результат в заданном диапазоне

```
main.cpp x fun1.S x
1  #include <stdio.h>
2
3  extern "C" int func (int a, int b)
4
5  {
6      printf("Function\'s params:\na:: %d \nb:: %d", a, b);
7  }
8
9
10

.data
    a: .int 13
    b: .int 21
.global __main
__main:
    push a
    push b
    call __func

    addl $2*4, %esp
    xorl %eax, %eax
    ret
```

```

H:\avsLab3\asm3\bin\Debug\asm3.exe
Function's params:
a:: 21
b:: 13
Process returned 0 (0x0)   execution time : 0.047 s
Press any key to continue.

```

**Задание 4. Бонус (+2 балла).** Опишите на ассемблере одну подпрограмму с параметрами  $a, b, \dots$  и результатами  $x$  и  $y$  и вызовите её из другой ассемблерной программы.

$(N^{\circ} - 1) \% 2 + 1$	Вариант
1	$\begin{cases} x = a + c \cdot b \\ y = a - c \cdot b \end{cases}$
2	$\begin{cases} x = a^2 - b^2 \\ y = 2ab \end{cases}$

```

1  .data
2      a: .int 4
3      b: .int 4
4      c: .int 5
5  .global _fun4
6  _fun4:
7      mov b, %eax
8      imul c, %eax
9      mov %eax, %ebp #bc
10     add a, %eax
11     mov %ebp, %ebx
12     neg %ebx
13     add a, %ebx
14     ret
15

```

```

1  .data
2      msg: .string "x=%d, y=%d\n"
3
4  .global _main
5  _main:
6      call _fun4
7      push %ebx
8      push %eax
9      push $msg
10     call _printf
11     addl $3*4, %esp
12     ret

```

```

x=24, y=-16

Process returned 12 (0xC)   execution time : 0.078 s
Press any key to continue.

```

