

**Министерство науки и образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
"Московский институт электронной техники"
(МИЭТ)**

Отчет по лабораторной работе № 4

" Модули и функции на ассемблере"

Выполнили: студенты ПМ - 31

Алтухова Анна Валерьевна

Мартынова Мария Олеговна

2021 г.

Задание Л4.№1. Разработайте ассемблерную функцию, вычисляющую целое выражение от двух целых аргументов (в соответствии с вариантом), а также головную программу на языке C/C++, использующую разработанную функцию.

(№ – 1)%5 +1	Вариант
1	$z(x, y) = 1234 + x - 5 \cdot y$

Листинг:

Ассемблерная функция:

```
.globl foo
foo:
sub $8, %rsp
mov %edx, %eax
imul $5,%eax
not %eax
inc %eax
add %ecx, %eax
add $1234,%eax
add $8, %rsp
ret
```

Программа на C++, запускающая ассемблерную функцию и проверка:

```
#include <iostream>
#include <locale.h>
using namespace std;
extern "C" unsigned foo(unsigned x, unsigned y);

int check(int x, int y)
{
    return 1234+x-5*y;
}

int main()
{
    setlocale(0, "");
    cout<<"-----Задание 1-----"<<endl;
    int x=1,y=0;
    cout <<"Решение с помощью модуля на ассемблере: z="<< foo(x, y) <<endl;
    cout<<"Проверка: z="<<check(x,y)<<endl;
    return 0;
}
```

Вывод:

```
-----Задание 1-----  
Решение с помощью модуля на ассемблере: z=1235  
Проверка: z=1235  
  
Process returned 0 (0x0)   execution time : 0.072 s  
Press any key to continue.
```

Задание Л4.№2. Разработайте программу, целиком написанную на ассемблере, вычисляющую (вызывая ранее разработанную функцию) значение $z(x, y)$ для $x = 1, y = 0$ и выводящую полученное значение на стандартный вывод с использованием библиотеки `libc` (в частности, функции `printf`).

Листинг:

```
.data
fmt: .string "-----Task 2-----\nResult: z=%d\n"
.text

.globl foo
foo:
sub $8, %rsp
mov %edx, %eax
imul $5, %eax
not %eax
inc %eax
add %ecx, %eax
add $1234, %eax
add $8, %rsp
ret

.globl main
main:

sub $32, %rsp
mov $1, %ecx
mov $0, %edx
call foo
add $32, %rsp

lea fmt(%rip), %rcx
mov %eax, %edx
sub $32, %rsp
call printf
add $32, %rsp

xor %eax, %eax
ret
```

Вывод:

```
-----Task 2-----
Result: z=1235

Process returned 0 (0x0)   execution time : 0.056 s
Press any key to continue.
```

Задание Л4.№3. Опишите на произвольном языке высокого уровня (включая C/C++) функцию с пятью целочисленными параметрами, которая выводит свои параметры на экран и возвращает результат, равный пятому параметру. Вызовите её из ассемблерной функции (в том числе из написанной на ассемблере *main()*).

Листинг:

Функция на C++:

```
#include <iostream>
#include <locale.h>
using namespace std;
extern "C" int fifth_arg(int arg1,int arg2,int arg3,int arg4,int arg5);
int fifth_arg(int arg1,int arg2,int arg3,int arg4,int arg5)
{
    setlocale(0, "");
    cout<<"Аргументы функции:"<<endl;
    cout<<"1 - "<<arg1<<endl;
    cout<<"2 - "<<arg2<<endl;
    cout<<"3 - "<<arg3<<endl;
    cout<<"4 - "<<arg4<<endl;
    cout<<"5 - "<<arg5<<endl;
    return arg5;
}
```

Её вызов из ассемблерного main:

```
.data
fmt1: .string "-----Task 3-----\n"
fmt2: .string "Result: arg5=%d\n"
.text

.globl main
main:

    lea fmt1(%rip), %rcx
    mov %eax, %edx
    sub $32, %rsp
    call printf
    add $32, %rsp

    mov $1, %ecx
    mov $2, %edx
    mov $3, %r8d
    mov $4, %r9d
    mov $5, %rbp
```

```
push %rbp
sub $32, %rsp
call fifth_arg
add $32, %rsp
pop %rbp

lea fmt2(%rip), %rcx
mov %eax, %edx
sub $32, %rsp
call printf
add $32, %rsp

xor %eax, %eax
ret
```

Вывод:

```
-----Task 3-----
Аргументы функции:
1 - 1
2 - 2
3 - 3
4 - 4
5 - 5
Result: arg5=5

Process returned 0 (0x0)   execution time : 0.086 s
Press any key to continue.
```

Задание Л4.№4. Бонус +2 балла для пар, обязательное для троек. Опишите на ассемблере одну подпрограмму с комплексным параметром $z = (z.re, z.im)$ и комплексным результатом $w = (w.re, w.im)$ (вещественные и мнимые части считать целочисленными) и вызовите её из другой ассемблерной программы.

$(N - 1) \% 2 + 1$	Вариант
1	$w = z^2 + (100, 200)$

Входные параметры: $Rz = x$ – запишем в регистр *ecx*

$Imz = y$ – запишем в регистр *edx*

Выходные параметры:

$Rw = x^2 + y^2 + 100$ – вычислим, результат запишем в регистр *edx* и выведем на экран

$Imz = 2xy + 200$ – вычислим, результат запишем в регистр *r8d* и выведем на экран

Листинг:

```
.globl Complex_func
Complex_func:
.data
fmt: .string "-----Task 4-----\nResult: w=(%d,%d)\n"
.text
sub $8, %rsp

mov %edx, %eax
imul %ecx, %edx
imul $2, %edx
add %eax, %ecx
imul %ecx, %ecx
sub %edx, %ecx
add $100, %ecx
add $200, %edx

mov %edx, %r8d
mov %ecx, %edx

lea fmt(%rip), %rcx
sub $32, %rsp
call printf
add $32, %rsp
```

```

add $8, %rsp
ret

.globl main
main:

    push %rax
    sub $32, %rsp
    mov $0, %ecx
    mov $0, %edx
    call Complex_func
    add $32, %rsp

    pop %rax
    xor %eax, %eax
    ret

```

Вывод:

```

-----Task 4-----
Result: w=(100,200)

Process returned 0 (0x0)   execution time : 0.177 s
Press any key to continue.

```