«Национальный исследовательский университет

«Московский институт электронной техники»

Лабораторная работа №3 по дисциплине «Архитектура вычислительных систем»

Модули и функции на ассемблере

Выполнили студены группы МП-25

Саядян Артём Грачикович

Калинкин Никита Анатольевич

Констандогло Александр Витальевич

**Москва 2017**

Вариант №2

**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.

Вариант: *y(x) = x%4 – x*

**Результат:**

**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 stdlib (в частности, функции printf).

**Результат:**

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

Вариант: ввод результата с клавиатуры.

**Результат:**

**GCC (x86\_64-win32-seh-rev0, Built by MinGW-W64 project) 7.2.0**

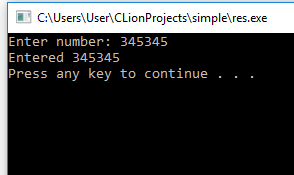
**ex3.cpp**

#include **<cstdio>**#include **<cstdlib>  
  
extern "C"  
void** ex3() {  
 **int** a;  
 printf(**"Enter number: "**);  
 scanf(**"%d"**, &a);  
 printf(**"Entered %d \n"**, a);  
 system(**"pause"**);  
}

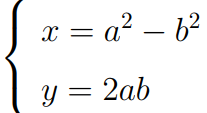
**ex3.s**

**.def** main  
main:  
 **subq** $40, **%**rsp  
 **call** ex3  
 **xorl %**eax, **%**eax  
 **addq** $40, **%**rsp  
 **ret**

**Компиляция:** *gcc ex3.cpp ex3.s -o ex3.exe*



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



**Результат:**

**GCC (x86\_64-win32-seh-rev0, Built by MinGW-W64 project) 7.2.0**

**ex4.s**

**.def** main  
main:  
 **pushq %**rbp  
 **movq %**rsp, **%**rbp  
 **subq** $32, **%**rsp  
 **movl** $25, **%**edx  
 **movl** $17, **%**ecx  
 **call** func  
 **movl** $0, **%**eax  
 **addq** $32, **%**rsp  
 **popq %**rbp  
 **ret**

**func.s**

x:  
 **pushq %**rbp  
 **movq %**rsp, **%**rbp  
 **movl %**ecx, 16(**%**rbp)  
 **movl %**edx, 24(**%**rbp)  
 **movl** $2, **%**eax  
 **subl** 24(**%**rbp), **%**eax  
 **xorl** 16(**%**rbp), **%**eax  
 **xorl** $2, **%**eax  
 **popq %**rbp  
 **ret**y:  
 **pushq %**rbp  
 **movq %**rsp, **%**rbp  
 **movl %**ecx, 16(**%**rbp)  
 **movl %**edx, 24(**%**rbp)  
 **movl** 16(**%**rbp), **%**eax  
 **addl %**eax, **%**eax  
 **imull** 24(**%**rbp), **%**eax  
 **popq %**rbp  
 **ret  
  
 .globl** func  
func:  
 **pushq %**rbp  
 **movq %**rsp, **%**rbp  
 **subq** $48, **%**rsp  
 **movl %**ecx, 16(**%**rbp)  
 **movl %**edx, 24(**%**rbp)  
 **movl** 24(**%**rbp), **%**eax  
 **movl %**eax, **%**edx  
 **movl** 16(**%**rbp), **%**ecx  
 **call** x  
 **movl %**eax, -8(**%**rbp)  
 **movl** 24(**%**rbp), **%**eax  
 **movl %**eax, **%**edx  
 **movl** 16(**%**rbp), **%**ecx  
 **call** y  
 **movl %**eax, -4(**%**rbp)  
 **movq** -8(**%**rbp), **%**rax  
 **addq** $48, **%**rsp  
 **popq %**rbp  
 **ret**