Федеральное государственное образовательное бюджетное учреждение  
высшего образования

«Финансовый университет при Правительстве Российской Федерации»

Колледж информатики и программирования

ОТЧЁТ

по практической работе № 3

**Дисциплина**: «Машинно ориентированное программирование для решения задач защиты информации»

Обучающегося (-ейся) Шатских Полина Сергеевна

Специальность:10.02.03 Обеспечение информационной безопасности автоматизированных систем

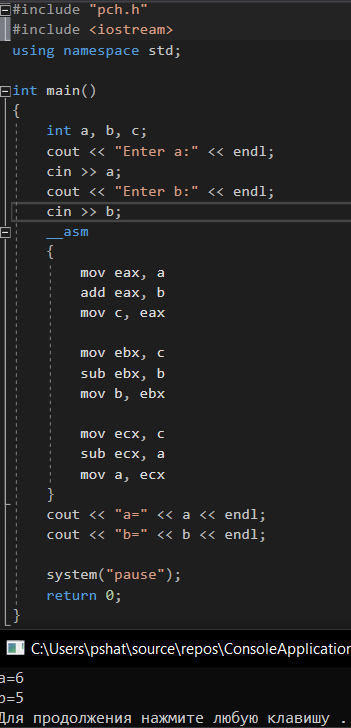
Группа: 3ОИБАС-718

Оценка: \_\_\_\_\_\_\_\_\_\_

Преподаватель: И.В. Сибирев

Москва, 2020

Задание 3.0:

1. Обменять значения в переменных языка Паскаль x : integer и y : ^integer.

Код программы:

#include "pch.h"

#include <iostream>

using namespace std;

int main()

{

int a, b, c;

cout << "Enter a:" << endl;

cin >> a;

cout<< "Enter b:" << endl;

cin >> b;

\_\_asm

{

mov eax, a

add eax, b

mov c, eax

mov ebx, c

sub ebx, b

mov b, ebx

mov ecx, c

sub ecx, a

mov a, ecx

}

cout << "a=" << a << endl;

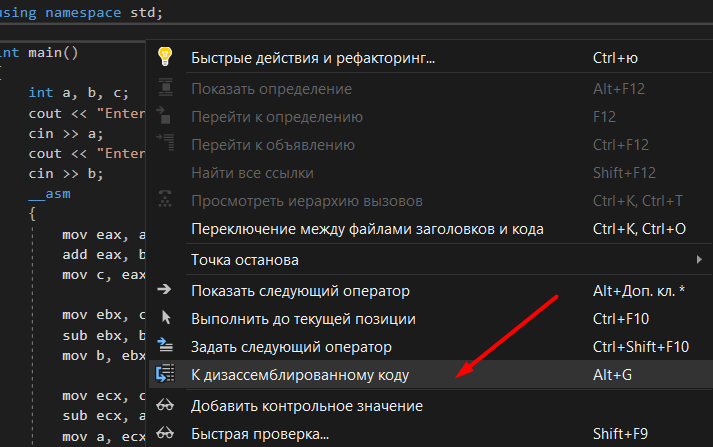
cout << "b=" << b << endl;

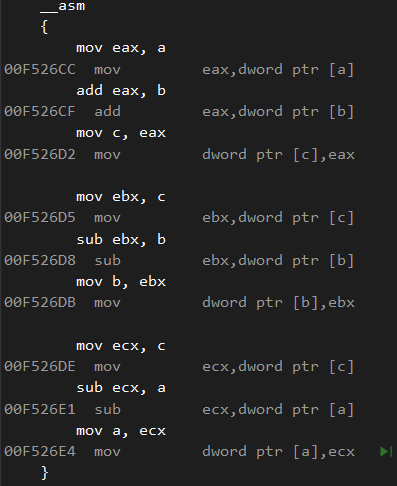
system("pause");

return 0;

}

2. Далее переходим к дизассемблированному коду. Нажимаем клавишу F10 и в контекстном меню выбираем «к дизассемблированному коду»

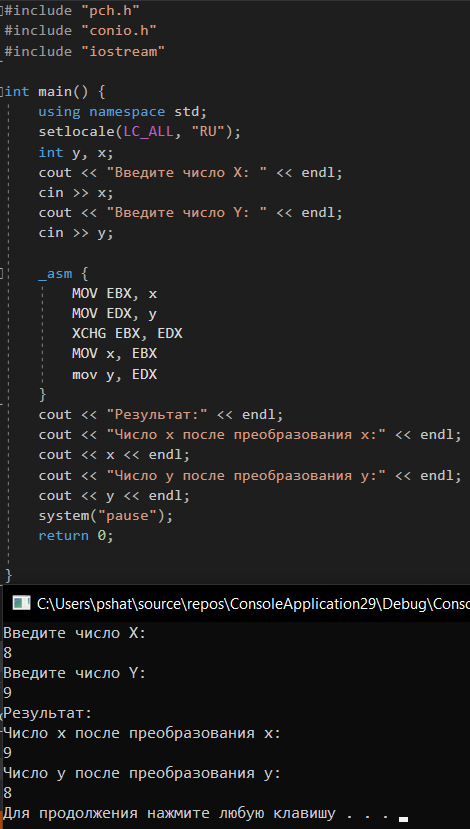




В окне Дизассемблированный код отображается код сборки, соответствующий инструкциям, созданным компилятором. При отладке управляемого кода эти инструкции ассемблера соответствуют присущему данному объекту коду, созданному компилятором.

Задание 3.1:

1. Обменять значения в переменных языка Паскаль x : integer и y : ^integer. Обязательно использовать команду XCHG.

Код программы:

#include "pch.h"

#include "conio.h"

#include "iostream"

int main() {

using namespace std;

setlocale(LC\_ALL, "RU");

int y, x;

cout << "Введите число X: " << endl;

cin >> x;

cout << "Введите число Y: " << endl;

cin >> y;

\_asm {

MOV EBX, x

MOV EDX, y

XCHG EBX,EDX

MOV x,EBX

mov y,EDX

}

cout << "Результат:" << endl;

cout << " Число x после преобразования x:" << endl;

cout << x << endl;

cout << "Число y после преобразования y:" << endl;

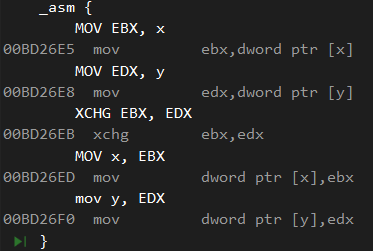
cout << y << endl;

system("pause");

return 0;

}

2. Переход к дизассемблированному коду.



В дополнение к инструкциям ассемблера в окне Дизассемблированный код могут отображаться следующие сведения:

* Адреса в памяти, где располагается каждая из инструкций. Для собственных приложений это фактические адреса в памяти.
* Исходный код, из которого получается код сборки.
* Байты кода, то есть байтовое представление реальных инструкций компьютера или языка MSIL.
* Символьные имена для адресов памяти.
* Номера строк, соответствующие исходному коду.