[15:55, 23.09.2020] Иван: https://www.sites.google.com/site/sistprogr/lekcii1/lek6

[16:06, 23.09.2020] Иван: // ConsoleApplication4.cpp : Этот файл содержит функцию "main". Здесь начинается и заканчивается выполнение программы.

//

#include <iostream>

void doSomething();

int main()

{

int x= 9;

int y = 8;

std::cout <<"x="<< x << std::endl;

std::cout << "y=" << y << std::endl;

int c = 0;

\_asm

{

mov eax, dword ptr[x]

mov dword ptr[c], eax

mov eax, dword ptr[y]

mov dword ptr[x], eax

mov eax, dword ptr[c]

mov dword ptr[y], eax

}

{

/\*

c = x;

x = y;

y = c;

\*/

}

std::cout << "x=" << x << std::endl;

std::cout << "y=" << y << std::endl;

if (false)

{

//оБЛАСТЬ ДЕЙСТВИЯ - ВЕСЬ ФАЙЛ, ИЛИ ВЕСЬ ПРОЕКТ - НУЖНО ПРОВЕРИТЬ

#define MY\_FAVORITE\_NUMBER 7

std::cout << "My favorite number is: " << MY\_FAVORITE\_NUMBER << std::endl;

#define MY\_FAVORITE\_VALUE "qwerty"

std::cout << "My favorite VALUE is: " << MY\_FAVORITE\_VALUE << std::endl;

#define USE\_YEN

std::cout << "My favorite USE\_YEN is: " << USE\_YEN 2<< std::endl;

#ifdef USE\_YEN

std::cout << "USE\_YEN - Define\n";

#endif

#ifndef USE\_YEN\_NOT

std::cout << "USE\_YEN\_NOT - NOT Define\n";

#endif

}

else if (!false)

{

#define PRINT

#undef PRINT

#ifdef PRINT

std::cout << "Printing!";

#endif

#ifndef PRINT

std::cout << "Not printing!";

#endif

#undef PRINT

#define PRINT

doSomething();

}

else if (!false)

{

}

else if (!false)

{

}

std::cout << "Hello World!\n";

}

void doSomething()

{

#ifdef PRINT

std::cout << "Printing!";

#endif

#ifndef PRINT

std::cout << "Not printing!";

#endif

}

// Запуск программы: CTRL+F5 или меню "Отладка" > "Запуск без отладки"

// Отладка программы: F5 или меню "Отладка" > "Запустить отладку"

// Советы по началу работы

// 1. В окне обозревателя решений можно добавлять файлы и управлять ими.

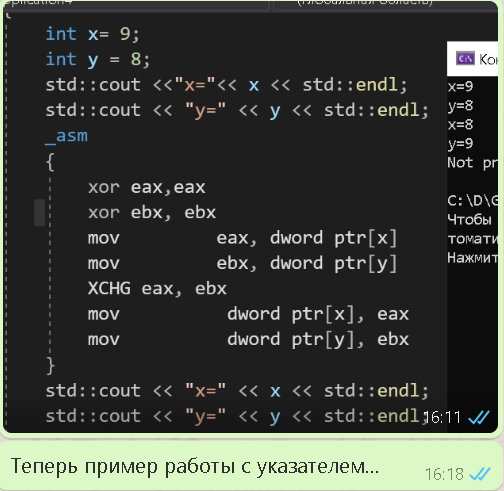
// 2. В окне Team Explorer можно подключиться к системе управления версиями.

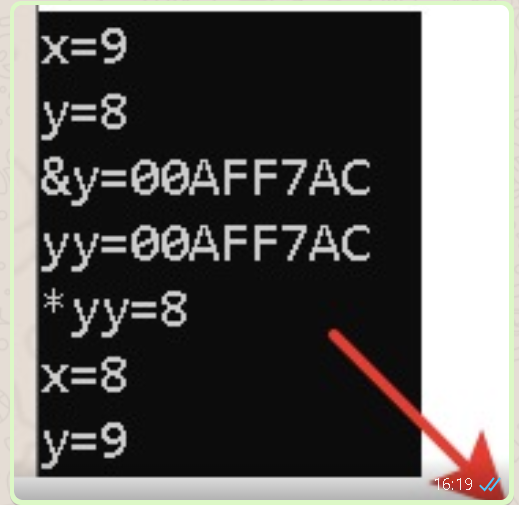
// 3. В окне "Выходные данные" можно просматривать выходные данные сборки и другие сообщения.

// 4. В окне "Список ошибок" можно просматривать ошибки.

// 5. Последовательно выберите пункты меню "Проект" > "Добавить новый элемент", чтобы создать файлы кода, или "Проект" > "Добавить существующий элемент", чтобы добавить в проект существующие файлы кода.

// 6. Чтобы снова открыть этот проект позже, выберите пункты меню "Файл" > "Открыть" > "Проект" и выберите SLN-файл.





//переременная x

int x= 9;

//переременная y

int y = 8;

//Указатель на пернеменную yy

int\* yy = &y;

std::cout <<"x="<< x << std::endl;

std::cout << "y=" << y << std::endl;

//Обращение к адресу оперативной памяти переменной

std::cout << "&y=" << &y << std::endl;

//Просто посмотреть что внутри указателя

std::cout << "yy=" << yy << std::endl;

//Обращение к значению по удресу из указателя

std::cout << "\*yy=" << \*yy << std::endl;

\_asm

{

xor eax,eax

xor ebx, ebx

mov eax, dword ptr[x]

mov ecx, dword ptr[yy]

mov ebx, dword ptr[ecx]

XCHG eax, ebx

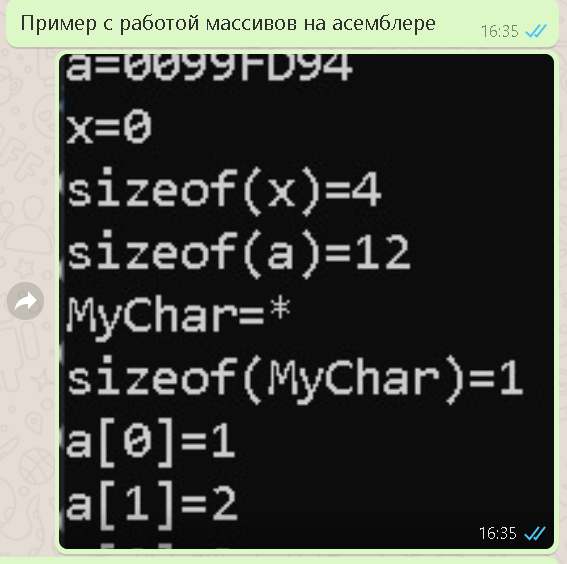
mov dword ptr[x], eax

mov dword ptr[ecx], ebx

}

std::cout << "x=" << x << std::endl;

std::cout << "y=" << y << std::endl;



//переременная x

int x= 0;

//Создали одномерный массив из 3 элементов

int a[3] = { 1, 2, 3};

//Создали символ

char MyChar = '\*';

//Вывели наш массив на экран в стиле аля питон

std::cout << "a={" << a[0]<<","<< a[1] << ","<< a[2] << ","<<"}"<< std::endl;

//Вывели на экран весь массив, и вдруг выяснилось что это указатель

std::cout << "a=" << a<< std::endl;

std::cout << "x=" << x << std::endl;

//Получили размер переменной x которая int

std::cout << "sizeof(x)="<< sizeof(x) << std::endl;//=4

//Получили размер всего нашего массива (он 12) из 3 элементов размером по 4

std::cout << "sizeof(a)=" << sizeof(a) << std::endl;

std::cout << "MyChar=" << MyChar << std::endl;

//Вывели размер символа

std::cout << "sizeof(MyChar)=" << sizeof(MyChar) << std::endl;

//Вывели на екран первый элемент массива, обратившись к нему через ассемблер

std::cout << "a[0]=";

\_asm

{

xor eax,eax

mov eax, dword ptr[a]

mov dword ptr[x], eax

}

std::cout << x << std::endl;

//Вывели на екран первый элемент массива, обратившись к нему через ассемблер

//НеЗАБЫЛИ У INT ПРО РАЗМЕР И ДОПИСАЛИ СМЕШЕНИЕ В ОПЕРАТИВНОЙ ПАМЯТИ +4

std::cout << "a[1]=";

\_asm

{

xor eax, eax

mov eax, dword ptr[a+4\*1]

mov dword ptr[x], eax

}

std::cout << x << std::endl;

//+4 2 раза

std::cout << "a[2]=";

\_asm

{

xor eax, eax

mov eax, dword ptr[a + 4 \* 2]

mov dword ptr[x], eax

}

std::cout << x << std::endl;

//Обратились к 4, несуществующему элементу массива

//Получили мусор

std::cout << "a[3]=";

\_asm

{

xor eax, eax

mov eax, dword ptr[a + 4 \* 3]

mov dword ptr[x], eax

}

std::cout << x << std::endl;

и не забудте : f10/Перейти к дезассемблированному коду/f10

=)