Национальный исследовательский университет «Московский институт электронной техники»

Лабораторная работа №2

По «Архитектуре вычислительных систем»

Выполнили: Зиновьева Е.

Ткаченко В.

Зеленоград

2021

Задание Л2.№1. Разработайте программу на языке C/C++, вычисляющую целое выражения от целого аргумента (в соответствии с вариантом). Переменные x, y, z — локальные переменные функции $main()$.

$(№ - 1)\%5 + 1$	**Вариант**
1	$z = f(x, y) = x + y + 177$
2	$z = f(x, y) = x - y - 512$
3	$z = f(x, y) = -x - y + 789$
4	$z = f(x, y) = -x + y - 348$
5	$z = f(x, y) = x + y - 451$

Примечание: Не обязательно вводить x и y с клавиатуры. Инициализация локальной переменной «узнаваемым» литералом упростит поиск её адреса.

Поставьте точку останова на начало кода, вычисляющего $f(x, y)$.

Установите для отображения дизассемблированного кода синтаксис AT&T.

Запустите программу, после останова переключитесь на уровень инструкций и изучите ассемблерный код, соответствующий вычислениям.

Определите и прокомментируйте (по возможности):

- обращение к переменным x, y и z;
- обращение к регистрам общего назначения;
- арифметические и логические операции.

Выполните код по шагам на уровне инструкций (выполняя за один шаг одну команду ассемблера).

Задание Л2.№2. Откройте окно «Регистры» и отследите изменение регистров общего назначения и регистра флагов при вычислении $f(x, y)$.

Задание Л2.№3. Откройте окно «Память» и отследите изменение переменных x, y и z. Сопоставьте адреса локальных нестатических переменных и значение регистра sp (указателя вершины стека: rsp в 64-битном режиме и esp в 32-битном).

Код программы:

```cpp
#include <iostream>

using namespace std;

int main()
{
    int x = 15;
    int y = 10;
    int z = -x + y - 348;
    cout << z;
}
```

```
00971810  push        ebp
00971811  mov         ebp,esp
00971813  sub         esp,0E4h
00971819  push        ebx
0097181A  push        esi
0097181B  push        edi
0097181C  lea         edi,[ebp-24h]
0097181F  mov         ecx,9
00971824  mov         eax,0CCCCCCCCh
00971829  rep stos    dword ptr es:[edi]
0097182B  mov         ecx,offset _F5AC55ED_Lab2@cpp (097C029h)
00971830  call        @__CheckForDebuggerJustMyCode@4 (0971311h)
```

Здесь показаны команды, вызывающиеся перед вызовом функции main. Далее видны

```
00971835  mov         dword ptr [x],0Fh
```

```
0097183C  mov         dword ptr [y],0Ah
```

команды который присваивают переменным значения в шестнадцатеричном формате, переменной x мы присваиваем значение 15, а переменной y - значение 10.

```
00971843  mov         eax,dword ptr [x]
00971846  neg         eax
00971848  mov         ecx,dword ptr [y]
0097184B  lea         edx,[eax+ecx-15Ch]
00971852  mov         dword ptr [z],edx
```

Далее видны команды которые вычисляют число для записи по адресу z.

1. Первая команда записывает в регистр eax значение по адресу переменной x.
2. Вторая команда отрицает его, т.е. делает негативным.
3. Третья команда записывает в регистр ecx значение по адресу переменной y.
4. Четвертая команда записывает в регистр edx значение выражения состоящее из суммы значений регистров eax и ecx минус число “348” в шестнадцатеричном представлении.
5. В пятой строке записываем по адресу, где лежит переменная z значение, вычисленное на предыдущем шаге.

На вкладке регистры можем отследить изменение значений регистров в ходе этих операций:

Начальное состояние

```
Registers
 EAX = 0097C029 EBX = 00785000
   ECX = 0097C029 EDX = 00000001
   ESI = 0095F6EC EDI = 0095F6E8
   EIP = 00971843 ESP = 0095F5F8
   EBP = 0095F6E8 EFL = 00000246

 0x0095F6E0 = 0000000F
```

Выполнили первую команду:

```
EAX = 0000000F EBX = 00785000
  ECX = 0097C029 EDX = 00000001
  ESI = 0095F6EC EDI = 0095F6E8
  EIP = 00971846 ESP = 0095F5F8
  EBP = 0095F6E8 EFL = 00000246
```

Выполнили вторую:

```
Registers
 EAX = FFFFFFF1 EBX = 00785000
   ECX = 0097C029 EDX = 00000001
   ESI = 0095F6EC EDI = 0095F6E8
   EIP = 00971848 ESP = 0095F5F8
   EBP = 0095F6E8 EFL = 00000293

 0x0095F6D4 = 0000000A
```

Выполнили третью:

```
Registers
 EAX = FFFFFFF1 EBX = 00785000
   ECX = 0000000A EDX = 00000001
   ESI = 0095F6EC EDI = 0095F6E8
   EIP = 0097184B ESP = 0095F5F8
   EBP = 0095F6E8 EFL = 00000293
```

Выполнили четвертую:

```
EAX = FFFFFFF1 EBX = 00785000
  ECX = 0000000A EDX = FFFFFE9F
  ESI = 0095F6EC EDI = 0095F6E8
  EIP = 00971852 ESP = 0095F5F8
  EBP = 0095F6E8 EFL = 00000293

0x0095F6C8 = CCCCCCCC
```

Выполнили пятый шаг:

```
Registers
 EAX = FFFFFFF1 EBX = 00785000
   ECX = 0000000A EDX = FFFFFE9F
   ESI = 0095F6EC EDI = 0095F6E8
   EIP = 00971855 ESP = 0095F5F8
   EBP = 0095F6E8 EFL = 00000293
```

Далее идет вызов системных функций для вывода значений z на экран, а также завершение работы функции main.

```
00971855  mov         esi,esp    ≤1ms elapsed
00971857  mov         eax,dword ptr [z]
0097185A  push        eax
0097185B  mov         ecx,dword ptr [__imp_std::cout (097B098h)]
00971861  call        dword ptr [__imp_std::basic_ostream<char,std::char_tra
00971867  cmp         esi,esp
00971869  call        __RTC_CheckEsp (097123Ah)
}
0097186E  xor         eax,eax
00971870  pop         edi
00971871  pop         esi
00971872  pop         ebx
00971873  add         esp,0E4h
00971879  cmp         ebp,esp
0097187B  call        __RTC_CheckEsp (097123Ah)
00971880  mov         esp,ebp
00971882  pop         ebp
00971883  ret
```