

Национальный исследовательский университет «Московский институт
электронной техники»

Лабораторная работа №3

По «Архитектуре вычислительных систем»

Выполнили: Зиновьева Е.

Ткаченко В.

Зеленоград

2021

Задание Л3.№1. Реализуйте расчёт беззнакового целочисленного выражения (таблица Л3.1) как ассемблерную вставку в программу на C/C++, помещая временные значения в регистры общего назначения A, C, D (в зависимости от разрядности данных — eax, ecx, edx или rax, rcx, rdx).

Переменные x, y, z, w — локальные переменные функции $main()$ (или другой функции C/C++) и передаются во вставку как параметры (z и w — выходные, x и y — входные). Проверьте расчёт, реализовав то же самое на C/C++.

Варианты выражений для расчёта

Таблица Л3.1

$(N - 1) \% 3 + 1$	Вариант
1	$\begin{cases} z = (x + 79) / y \\ w = (x + 79) \% y \end{cases}$
2	$\begin{cases} z = x / (y - 134) \\ w = x \% (y - 134) \end{cases}$
3	$\begin{cases} z = (x + 8) / y \\ w = (x + 8) \% y \end{cases}$

Код:

```

2 void AssemblerFunc(int x, int y)
3 {
4     int z, w;
5     __asm
6     {
7         mov     eax, dword ptr[x] // записали в регистр eax значение переменной x
8         add     eax, 8             // добавили к этому значению 8
9         cdq     // расширяем размерность числа с учетом знака
10        div     dword ptr[y]       // делим на переменную y, записывая остаток в edx
11        mov     dword ptr[z], eax // поместили в z значение (x + 8) / y
12    }
13
14    __asm
15    {
16        mov     eax, dword ptr[x] // записали в регистр eax значение переменной x
17        add     eax, 8             // добавили к этому значению 8
18        cdq     // расширяем размерность числа с учетом знака
19        div     dword ptr[y]       // делим на переменную y, записывая остаток в edx
20        mov     dword ptr[w], edx // поместили в w значение остаток от (x + 8) / y
21    }
22
23    std::cout << "z = " << z << std::endl << "w = " << w << std::endl;
24 }

```

Результат работы + проверка:

```

Microsoft Visual Studio
z = 3
w = 1
Check:
z = 3
w = 1

```

Как видно результаты совпали.

Задание Л3.№2. Реализуйте задание [Л3.№1](#), передав вставку в качестве входных параметров не значения x и y , а указатели $p = \&x$ и $q = \&y$.

Код:

```

7 void AssemblerAddressFunc(int* x, int* y)
8 {
9     int z, w;
10    __asm
11    {
12        mov     eax, dword ptr[x]    // записали в регистр eax значение адреса переменной x
13        mov     eax, dword ptr[eax]  // записываем в регистр eax значение переменной x по адресу
14        add     eax, 8               // добавили к этому значению 8
15        mov     ecx, dword ptr[y]    // записали в регистр ecx адрес переменной y
16        cdq                     // расширяем размерность числа с учетом знака
17        div     dword ptr[ecx]       // делим на переменную y, записывая остаток в edx
18        mov     dword ptr[z], eax    // поместили в z значение (x + 8) / y
19    }
20
21    __asm
22    {
23        mov     eax, dword ptr[x]    // записали в регистр eax значение адреса переменной x
24        mov     eax, dword ptr[eax]  // записываем в регистр eax значение переменной x по адресу
25        add     eax, 8               // добавили к этому значению 8
26        mov     ecx, dword ptr[y]    // записали в регистр ecx адрес переменной y
27        cdq                     // расширяем размерность числа с учетом знака
28        div     dword ptr[ecx]       // делим на переменную y, записывая остаток в edx
29        mov     dword ptr[w], edx    // поместили в w значение остатка от (x + 8) / y
30    }
31
32    std::cout << "z = " << z << std::endl << "w = " << w << std::endl;
33 }

```

Результат работы + проверка:

```

Microsoft Visual Studio
z = 3
w = 1
Check:
z = 3
w = 1

```

Как видно результаты совпали.

Задание Л3.№4. На языке C/C++ выделите память под массив M (статический или динамический) из N целых чисел и инициализируйте M нулями.

Реализуйте для заданного $k \in [0, N)$ запись значения $x \neq 0$ на место элемента $M[k]$, используя компоненты эффективного адреса ($Base, Index, 2^{Scale}$).

Код:

```
std::cout << "Create array: " << std::endl;

int n = 5;
int* pArray = new int[n];

InitializeArray(pArray, n);
SetNumber(pArray, 1, 789);
//SetFF(pArray, 1);

delete[] pArray;
```

```
void InitializeArray(int pArray[], size_t n)
{
    __asm
    {
        mov ecx, n // записываю в регистр ecx сколько раз ходить по циклу
        mov esi, 0
        mov edx, 0 // ход конем, так как просто 0 он сразу писать не хотел
        mov eax, pArray
        GO :
        mov[ecx + 4 * esi], edx // Заполняю нулями в цикле
        inc esi
        loop GO
    }

    for (size_t i = 0; i < n; i++)
    {
        std::cout << pArray[i] << " ";
    }
}

void SetNumber(int pArray[], unsigned int index, int number)
{
    __asm
    {
        mov edx, dword ptr[index] // записываем в ebx индекс
        mov ebx, dword ptr[number] // записываем в ebx number
        mov eax, pArray

        mov[ecx + 4 * edx], ebx
    }

    std::cout << std::endl << "New element on index " << index << " are equal " << pArray[index];
}
```

Результат:

```
Create array:  
0 0 0 0 0  
New element on index 1 are equal 789
```