

Лабораторная работа № 4:
“Использование ассемблерных
вставок в программах на С++”

МИЭТ

Выполнили:
студенты группы МП-30
Алимагадов К. А., Карпухин Г. К.
Вариант № 9

Зеленоград 2018 год

Задание № 1

Разработайте и выполните программу, в которой реализуйте при помощи ассемблерной вставки:

Вариант 3

Вычитание целых чисел без знака.

После команды сложения/вычитания проанализируйте соответствующий флаг состояния процессора, и, в зависимости от его значения, выведите результат и строку «Результат верный» (если не было переполнения) или «Результат неверный» (при переполнении).

```
#include <iostream>
using namespace std;
int main()
{
    cout << "Number 1\n";
    unsigned int p1;
    unsigned int p2;
    cout << "a - b = ?\n";
    cout << "Enter a:\n";
    cin >> p1;
    cout << "Enter b:\n";
    cin >> p2;

    asm                                     //Расширенная форма вставки с выходными
    параметрами
    (
        "subl %[b], %[a]\n\t"
        : [a] "+r" (p1) //Выходной параметр a используется для
        чтения и записи, и размещается в регистре
        : [b] "r" (p2) //Входной параметр b
        размещается в регистре
        : "cc" //Код изменяет регистр
        флагов
    );

    asm goto                               //Расширенная форма с метками
    выхода
    (
        "jc %l2" //Если флаг переполнения для
        беззнаковых чисел = 1, то перейти к метке carry
        :: [a] "r" (p1), [b] "r" (p2)
        : "cc"
        : carry //Метка carry является
        параметром с номером 2
    );
```

```

        cout << "\nTrue " << p1 << endl;

        if(0)
        {
            carry:
                cout << "\nFalse " << p1 << endl;
        }
    }
}

```

Задание №2

Реализуйте задание лабораторной работы ЛЗ (см. раздел ЛЗ.1, задание 1) как ассемблерную вставку в программу на C++.

$$y(x) = x^3$$

Задание №3

Реализуйте задание 2, не используя в тексте вставки конкретных имён регистров.

```

int main()
{
    int p = 0;
    cout << "x = ?\n";
    cin >> p;
    asm
    (
        "imull %[x], %[x]\n\t"
        "imull %[y], %[x], \n\t"
        : [x] "+r" (p)
        : [y] "r" (p)
        : "cc"
    );
    cout << "\n" << "x^3 = " << p;
    return 0;
}

```

Задание №4. Бонус (+2 балла). Придумайте и запишите в отчёте:

– маску (т.е. второй операнд бинарной побитовой операции) для преобразования кодов ASCII [59] цифр '0'...'9' в их двоичные эквиваленты (однобайтовые числа 0...9) с помощью операции AND. При этом ASCII-код нуля (шестнадцатеричное число 0x30, что также иногда записывается как 30₁₆, однако наиболее предпочтительное и самодокументирующееся представление ASCII-кода нуля — '0') должен быть преобразован в однобайтовое число 00, ASCII-код единицы ('1', шестнадцатеричное число 0x31 или 31₁₆) — в 01, и т. д.;

– необходимую операцию и маску для преобразования
однобайтовых чисел 0...9 в их коды ASCII;

– маску, которая преобразовывает строчные английские буквы в
прописные и наоборот с помощью операции XOR. Коды ASCII
приведены в приложении Б.

Реализуйте преобразования ASCII-кодов цифр в значения и
обратно как ассемблерную вставку в программу на C++.

```
int main()
{
    int Letter = 'A';
    int letter = 'a';
    int asciiicode = '9';
    int a = 9;
    int mask1 = 48; // = 1100 1111
    int mask2 = 207; // = 0011 0000
    int mask3 = 32; // = 0010 0000
    int result1, result2, result3, result4;
    cout << "Example = " << a << endl;
    cout << "ASCII CODE " << bitset<16>(asciiicode) << endl;
    cout << "Big letter = " << (char)Letter << endl;
    cout << "Small letter " << (char)letter << endl;

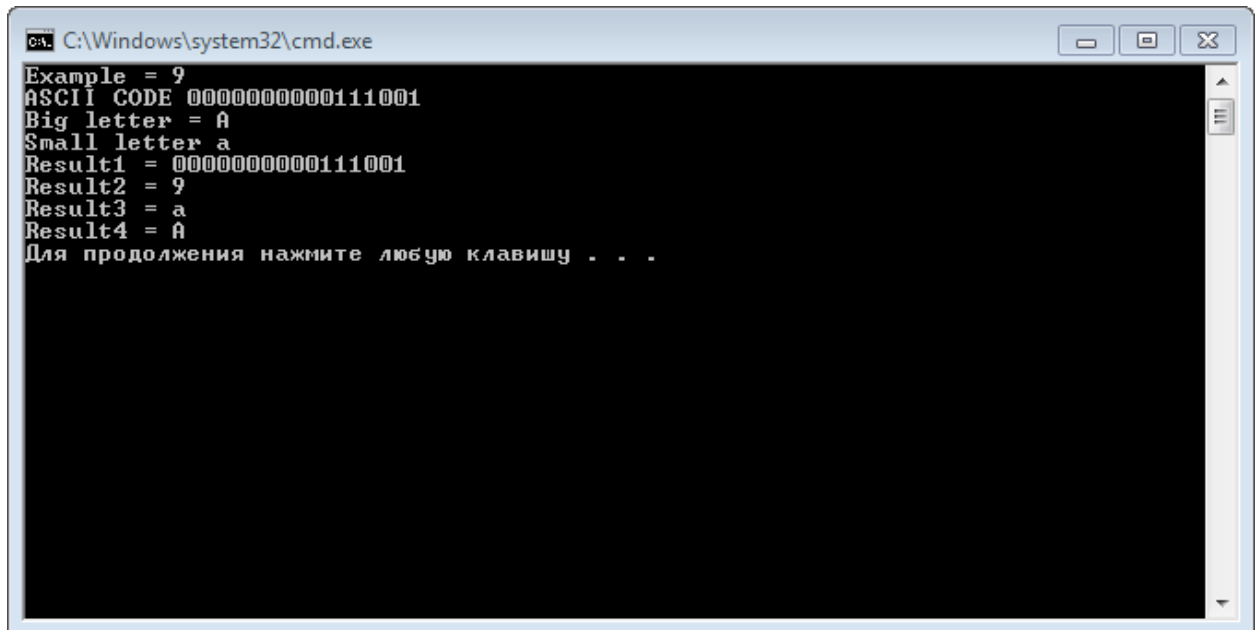
    _asm
    {
        mov eax, a // преобразование из числа в код ASCII
        or eax, mask1
        mov result1, eax

        mov eax, asciiicode // преобразование из кода ASCII в число
        and eax, mask2
        mov result2, eax

        mov eax, Letter // преобразование из прописной буквы в
        строчную
        xor eax, mask3
        mov result3, eax

        mov eax, letter // преобразование из строчной буквы в
        прописную
        xor eax, mask3
        mov result4, eax
    }
    cout << "Result1 = " << bitset<16>(result1) << endl;
    cout << "Result2 = " << result2 << endl;
    cout << "Result3 = " << (char)result3 << endl;
```

```
    cout << "Result4 = " << (char)result4 << endl;  
    return 0;  
}
```



A screenshot of a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The window has a black background with white text. The output of a program is displayed as follows:

```
Example = 9  
ASCII CODE 000000000000111001  
Big letter = A  
Small letter a  
Result1 = 000000000000111001  
Result2 = 9  
Result3 = a  
Result4 = A  
Для продолжения нажмите любую клавишу . . .
```

The text is aligned to the left. The last line is a Russian instruction to press any key to continue.