

# Лабораторная работа 4.

## Использование ассемблерных вставок в программах на C++

**Цель работы:** научиться вставлять в программы на языке высокого уровня ассемблерные фрагменты.

### Л4.1. Задание на лабораторную работу

**Задание 1.** Разработайте и выполните программу, в которой реализуйте при помощи ассемблерной вставки.

**Вычитание целых чисел без знака**

```
#include <stdio.h>

int main(void)
{
    unsigned int x = 3;
    unsigned int y = 1;
    bool flag = false;
    printf("x = %d\n y = %d\n", x, y);
    asm ( "sub %[Src], %[Dest]\n\t"
        : [Dest]"+r" (x)
        : [Src]"r" (y)
        : "cc" );
    asm ( "jno nooverflow;\n\
        "mov $0x1,%0;\n\
        "nooverflow :"\
        : "=r" (flag) : "0" (flag) );
    printf("x-y = %d\n", x);

    if (flag==true)
    {
        printf("Incorrect result\n");
    }
    else
    {
        printf("Correct result\n");
    }
    return 0;
}
```

```
x = 3
y = 1
x-y = 2
Correct result
```

**Задание 2.** Реализуйте задание лабораторной работы Л3

(см. раздел Л3.1, задание 1) как ассемблерную вставку в программу на C++.

**Задание 3. Бонус (+1 балл).** Реализуйте задание 2, не используя в тексте вставки конкретных имён регистров.

```
#include <stdio.h>

int main()
{
    int x = 20;
    int y;
    asm ( "imul $3, %0" : "=r" (y) : "0" (x) );
    asm ( "add $1, %0" : "=r" (y) : "0" (y) );
    printf("3*x+1 = %d", y);
}
```

**3\*x+1 = 61**

**Задание 4. Бонус (+2 балла).** Придумайте и запишите в отчёте:

Маску (т.е. второй операнд бинарной побитовой операции) для преобразования кодов ASCII 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++.

```
#include <iostream>
#include <stdlib.h>

using namespace std;

int main()
{
    unsigned int ASCIINumber = 0x39;
    unsigned int result;
    asm ( "and $0x0F, %0" : "=r" (result) : "0" (ASCIINumber) );
    cout<<result<<endl;
```

```
unsigned int numer = 0x09;
asm ( "xor $0x21, %0" : "=r" (ASCIINumber) : "0" (numer) );
cout<<ASCIINumber<<endl;

    return 0;
}
```

### Вопрос:

1. Каким ключевым словом открывается ассемблерная вставка?

asm

2. Как из ассемблерной вставки обратиться к локальным переменным?

```
asm ( "imul $-4, %0" : "=r" (y) : "0" (x) );
```

3. Какие вы знаете ограничения на размещение параметров ассемблерных вставок?

Общее количество параметров ограничено:  $input + output + goto \leq 30$