

Лабораторная работа №4
ПИН-31
(Краснов, Колпаков, Шокоров)

Задание 1:

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

| | |
|---|---------------------------------|
| 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>
#include <bitset>

using namespace std;
int main() {
    uint a, b, res = 0, flags;
    cout << "a = ";
    cin >> a;
    cout << "b = ";
    cin >> b;
    asm("mov %[A], %%eax\n"
        "sub %[B], %%eax\n"
        "mov %%eax, %[RES]\n"
        "lahf\n"
        "mov %%eax, %[FLAGS]"
        : [ RES ] "=r"(res), [ FLAGS ] "=r"(flags)
        : [ A ] "g"(a), [ B ] "g"(b)
        : "cc", "%eax" );
    /**
     * После lahf все флаги кладутся в регистр ah
     *
     * |-----ah-----|
     * A - [_____] [__ah__] [__al__]
     * |-----eax-----|
     *
     * Значит ah - второй байт справа
     * Разряды регистра ah и флаги, которые в них загружает команда lahf:
     *
     * 7   6   5   4   3   2   1   0
     * SF  ZF  0   AF  0   PF  1   CF
```

```

    * Значит нам нужен 0-ой разряд. Чтобы получить его из 32 бит нужный
    флаг,
    * достаточно просто выполнить операцию flags & 0b100000000 или flags &
    0x100
    */
    if ((flags & 0x100) == 0) {
        cout << "The result of [" << a << " - " << b << " = " << res <<"] is
correct" << endl;
    }
    else {
        cout << "The result of [" << a << " - " << b << " = " << res <<"] is
NOT correct, overflow detected!" << endl;
    }
    return 0;
}

```

```

[alexandr@mi-pro ~/MyFolder/Uni/Arch/lab4]$ ./task1
a = 5
b = 2
The result of [5 - 2 = 3] is correct
[alexandr@mi-pro ~/MyFolder/Uni/Arch/lab4]$ ./task1
a = 1
b = 7
The result of [1 - 7 = 4294967290] is NOT correct, overflow detected!
[alexandr@mi-pro ~/MyFolder/Uni/Arch/lab4]$ ./task1
a = -1
b = 7
The result of [4294967295 - 7 = 4294967288] is correct
[alexandr@mi-pro ~/MyFolder/Uni/Arch/lab4]$ ./task1
a = 5
b = -1
The result of [5 - 4294967295 = 6] is NOT correct, overflow detected!
[alexandr@mi-pro ~/MyFolder/Uni/Arch/lab4]$

```

Задание 2:

Задание 2. Реализуйте задание лабораторной работы Л3 (см. раздел Л3.1, задание 1) как ассемблерную вставку в программу на C++.

| | |
|---|-----------------|
| 3 | $y(x) = 3x + 1$ |
|---|-----------------|

```

#include <stdio.h>

int main()
{

```

```

int x = 4, y;

asm (
    "movl $3, %%ebx\n"
    "imul %[X], %%ebx\n"
    "inc %%ebx\n"
    "movl %%ebx, %[Y] "
    : [Y] "=r" (y)
    : [X] "r" (x)
    : "%ebx"
);

printf("%d", y);
return 0;
}

```

Задание 3:

Задание 3. Реализуйте задание 2, не используя в тексте вставки конкретных имён регистров.

```

#include <stdio.h>

int main()
{
    int x = 4, y, tmp;

    asm (
        "movl $3, %[TMP]\n"
        "imul %[X], %[TMP]\n"
        "inc %[TMP]\n"
        "movl %[TMP], %[Y] "
        : [Y] "=g" (y), [TMP] "=r" (tmp)
        : [X] "g" (x)
    );

    printf("%d", y);
    return 0;
}

```

Задание 4:

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++.

Маски:

From int -> char:

Маска = 0x0F (00001111) в совокупности с операцией AND.

From char -> int:

Маска = 0x30 (00110000) в совокупности с операцией OR.

From lower -> upper case:

Маска = 0x20 (00100000).

From int to char:

```
#include <iostream>

int main() {

    char ch;
    int number;
    std::cout << "Input a number: ";
    std::cin >> number;
    asm("mov %[NUMBER], %%eax\n\t"
        "or $0x30, %%al\n\t"
        "mov %%al, %[CHAR]\n\t"
        : [CHAR] "=m" (ch)
        : [NUMBER] "g" (number)
        );

    std::cout << ch << std::endl;
```

```

return 0;
}

(master)kdator@kdator:~/github/arch_vs/lab3$ ./test
Input a number: 1
1
(master)kdator@kdator:~/github/arch_vs/lab3$ ./test
Input a number: 2
2
(master)kdator@kdator:~/github/arch_vs/lab3$ ./test
Input a number: 3
3
(master)kdator@kdator:~/github/arch_vs/lab3$ ./test
Input a number: 9
9
(master)kdator@kdator:~/github/arch_vs/lab3$ ./test
Input a number: 10
:

```

По тестам видно, что он преобразовывает число в ASCII символы.

From char to int:

```

#include <iostream>

int main() {

    char ch;
    int number;
    std::cout << "Input a char: ";
    std::cin >> ch;
    asm("mov %[CHAR], %%al\n\t"
        "and $0x0F, %%al\n\t"
        "mov %%eax, %[NUMBER]\n\t"
        : [NUMBER] "=m" (number)
        : [CHAR] "g" (ch)
        );

    std::cout << number << std::endl;

    return 0;
}

```

```
(master)kdator@kdator:~/github/arch_vs/lab3$ ./test
Input a char: 0
0
(master)kdator@kdator:~/github/arch_vs/lab3$ ./test
Input a char: 1
1
(master)kdator@kdator:~/github/arch_vs/lab3$ ./test
Input a char: 6
6
(master)kdator@kdator:~/github/arch_vs/lab3$ ./test
Input a char: 9
9
(master)kdator@kdator:~/github/arch_vs/lab3$ ./test
Input a char: :
10
```