

ЛАБОРАТОРНАЯ РАБОТА №5

по дисциплине "Языки Ассемблера"

Выполнил Ковалев Даниил
СКБ171, вариант 12
МИЭМ НИУ ВШЭ

Дата: / / 2020

Баллы:

Задание: в массиве слов (байтов) указанного размера над каждым элементом выполнить операцию: если старший бит совпадает с младшим, то скопировать битовое поле 6:5 в поле 4:3.

```

1  #include <iostream>
2  #include <iomanip>
3  #include <vector>
4  #include <string>
5
6  using BYTE = std::int8_t;
7  using WORD = std::int16_t;
8
9  void lab_byte(BYTE* A)
10 {
11     asm( ".intel_syntax\n\t"
12         "mov rdi, rax\n\t" // адрес A в RDI
13         "mov ecx, 16\n\t"
14         "start_b:\n\t" // начало цикла
15         "mov al, [rdi]\n\t" // байт из массива A в AL
16         "mov dl, al\n\t" // байт из массива A в DL
17         "shr al, 7\n\t" // старший бит в AL
18         "and dl, 1\n\t" // младший бит в DL
19         "cmp al, dl\n\t"
20         "jne after_copy_b\n\t" // если ст. и мл. биты не равны, ничего не делать
21         "mov al, [rdi]\n\t" // байт из массива A в AL
22         "mov dl, al\n\t" // байт из массива A в DL
23         "and al, 231\n\t" // обнуляем биты 4:3
24         "and dl, 96\n\t" // обнуляем биты, кроме 6:5
25         "shr dl, 2\n\t" // кладем биты 6:5 в биты 4:3
26         "or al, dl\n\t"
27         "mov [rdi], al\n\t" // изменяем элемент массива
28         "after_copy_b:\n\t"
29         "inc rdi\n\t" // переходим к следующему элементу
30         "loop start_b\n\t"
31         :
32         : "a"(A)
33     );
34 }
35
36 void lab_word(WORD* A)
37 {
38     asm( ".intel_syntax\n\t"
39         "mov rdi, rax\n\t" // адрес A в RDI
40         "mov ecx, 16\n\t"
41         "start_w:\n\t" // начало цикла
42         "mov ax, [rdi]\n\t" // слово из массива A в AX
43         "mov dx, ax\n\t" // слово из массива A в DX
44         "shr ax, 15\n\t" // старший бит в AX
45         "and dx, 1\n\t" // младший бит в DX
46         "cmp ax, dx\n\t"
47         "jne after_copy_w\n\t" // если ст. и мл. биты не равны, ничего не делать
48         "mov ax, [rdi]\n\t" // слово из массива A в AX
49         "mov dx, ax\n\t" // слово из массива A в DX
50         "and ax, 65511\n\t" // обнуляем биты 4:3
51         "and dx, 96\n\t" // обнуляем биты, кроме 6:5
52         "shr dx, 2\n\t" // кладем биты 6:5 в биты 4:3
53         "or ax, dx\n\t"
54         "mov [rdi], ax\n\t" // изменяем элемент массива
55         "after_copy_w:\n\t"
56         "add rdi, 2\n\t" // переходим к следующему элементу
57         "loop start_w\n\t"
58         :
59         : "a"(A)
60     );
61 }

```

```

62
63 void print_words(const std::vector<WORD>& A)
64 {
65     std::cout << "{";
66     for (WORD elem : A)
67         std::cout << std::hex << std::setw(6) << (elem) << ' ';
68     std::cout << "}\n";
69 }
70
71 void print_bytes(const std::vector<BYTE>& A)
72 {
73     std::cout << "{";
74     for (BYTE elem : A)
75         std::cout << std::hex << std::setw(6) << (static_cast<int>(elem) & 255) << ' ';
76     std::cout << "}\n";
77 }
78
79 int main()
80 {
81     {
82         std::cout << std::string(55, ' ') + "BYTE\n";
83         std::vector<BYTE> A = { 105, 70, 90, 82, -114, 45, 59, 82, -13,
84                                35, -123, -70, 57, -98, 81, -32 };
85
86         std::cout << "Before:\n";
87         print_bytes(A);
88         lab_byte(A.data());
89         std::cout << "After:\n";
90         print_bytes(A);
91     }
92     {
93         std::cout << "\n" + std::string(55, ' ') + "WORD\n";
94         std::vector<WORD> A = { 4996, -28469, 12215, -14279, -22535, -16077, -16993,
95                                -26883, 510, 4745, -4753, 3704, -30365, 30061, 31930, 22819 };
96         std::vector<WORD> B = { 25757, -20781, -4543, 31013, -32147, -2348, 22367,
97                                27115, 30568, 6958, 25501, 23278, -25810, 22947, 4479, 1446 };
98
99         std::cout << "Before:\n";
100        print_words(A);
101        lab_word(A.data());
102        std::cout << "After:\n";
103        print_words(A);
104    }
105 }

```

