Тема: Изучение команд арифметических и логических операций

Цель: изучение команд организации циклов и способов косвенной адресации данных памяти в микропроцессоре i486; приобретение практических навыков составления программ обработки одномерных массивов, освоение методов анализа трудоемкости и ресурсной сложности алгоритмов обработки одномерных числовых массивов.

**Задание:**

1. На основании индивидуального задания «Дан массив В[16]. Определить среднее арифметическое массива, не используя команду деления» составить программу для обработки элементов одномерного массива. Длина элементов исходного массива равна DW. Значения элементов исходного массива задать в сегменте данных.

2. Получить загрузочный модуль и протестировать выполнение программы.

3. Выполнить расчет времени выполнения программы.

**Ход работы:**

1. Получим следующий код:

.686

include C:\masm32\include\io.asm

.data

;B dw 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11,

; 12, 13, 14, 15, 16

B dw 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 65,

70, 75, 80, 85, 91

Res dw 0

.code

start:

MOV AX, 0

MOV EBX, offset B

MOV ECX, 16

loop\_start:

mov DX, [EBX]

ADD AX, DX

ADD BX,2

LOOP loop\_start

del\_smech:

SHR AX, 4;

MOV Res, AX

exit

end start

В данной программе изначально обнуляется регистр AX, в EBX помещается значение, располагающееся по адресу массива, и задается счетчик на 16 элементов. В loop\_start происходит перебор элементов массива с вычислением их суммы (записывается в AX), в del\_smech – сдвиг вправо на 4 элемента. SHR выполняет в данном коде роль команды «деления», в связи с заданным условием задачи. Целая часть среднего арифметического помещается в Res.

2. Скомпилируем и запустим программу для значений от 1 до 16, получив следующие результаты:

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 1 – память на момент начала работы программы.

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

Рисунок 2 – Регистр EAX после подсчета суммы.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание



Рисунок 3– память на момент окончания работы программы.

Изменим входные данные массива А на значения с 43 по 84, и проверим работу программы при них.

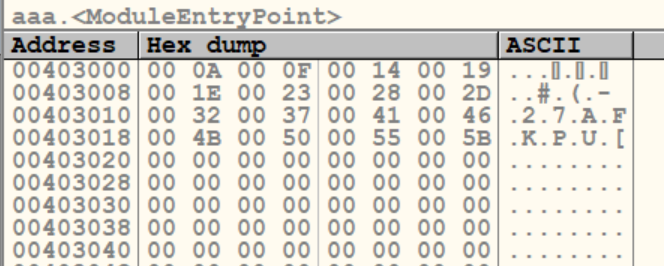


Рисунок 4 - память на момент начала работы программы.

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

Рисунок 5 – Регистр EAX суммы элементов массива.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание



Рисунок 6 - память на момент окончания работы программы.

3. Посчитаем время выполнения созданной программы:

1. MOV AX (аккумулятор), 0 (непосредственный операнд) – 4 такта
2. MOV ECX(регистр), 16 (операнд) – 4 байта
3. MOV DX, [EBX] -2 такта
4. ADD AX (регистр), DX(регистр) – 3 такта
5. ADD BX (регистр), 2 (операнд) – 4 такта
6. SHR AX (регистр), 4 (операнд) – 2 такта
7. MOV Res (память), AX (регистр) – 13(слова)+5(база) = 19 тактов

Общее время выполнения программы: 38 тактов.

**Результаты:**

В пункте 1 была проделана работа по созданию кода программы для вычисления арифметического выражения при помощи команд арифметических операций.

В задании 2 была скомпилирована, выполнена и продемонстрирована работа созданной и описанной в п.1.

В задании 3 был проведён расчёт времени выполнения созданного кода.

Вывод: были изучены работы команд арифметических и логических операций и написана программа для выполнения заданий, а также проведён анализ времени работы созданной программы.