ГУАП

КАФЕДРА № 34

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

| Старший преподаватель |  |  |  | Жиданов К.А. |
| --- | --- | --- | --- | --- |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ |
| --- |
|  |
| по курсу: |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

| СТУДЕНТ ГР. | 3145 |  |  |  | Скрипников Я.А. |
| --- | --- | --- | --- | --- | --- |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2021

**Задание 1**

Написать программу, реализующую заданный алгоритм: вычислить сумму ненулевых элементов в массиве. Проверить корректность работы программы с помощью отладчика **Ход работы**

Чтобы определить сумму ненулевых элементов массива, необходимо обработать каждый элемент массива и сравнить его с нулем

Далее суммировать ненулевые значения и завершить программу

Например, если исходный массив состоит из чисел (0, 5, 3, 2, 0, 8, 2, 2), то сумма его ненулевых значений равна 22

Выбрал регистры для хранения входных и выходных значений: EBX - массив, DX - сумма ненулевых элементов

Занес программу в отладчик

Выполнила программу, поочерёдно подавая на вход тестовые наборы, после каждого запуска убеждаясь, что результат программы совпадает с вычисленным вручную занес результаты вычислений в отчет

**Код программы:**

%include "io.inc"

section .text

global CMAIN

CMAIN:

;write your code here

xor eax, eax

XOR EDX,EDX

LEA ESI, [a]

MOV ECX, 8

next:

MOV AL,[ESI]

CMP AL, 0

JNZ summ

next1:

INC ESI

LOOP next

JMP quit

summ:

INC DL

JMP next1

quit:

PRINT\_UDEC 1,DL

ret

section .data

a: DB 0, 5, 3, 2, 0, 8, 2, 2

**Задание 2:** написать программу, реализующую доступ к упакованному массиву с заданной разрядностью элементов: 6 бит

**Ход выполнения:**

1. Выбираем массив, элементы которого не превышают 63, например [**15, 21, 7, 12, 59, 30, 25, 11**]
2. Переводим числа в двоичную систему, а затем записываем биты от младшего к старшему.
3. Объединяем числа по 8 элементов и переводим в 16-ричную СС
4. Регистры для входных и выходных значений:

ESI – Адрес массива

AL, Bl – Элементы массива

1. Заносим программу в отладчик
2. Выполняем программу, проверяя каждый элемент поочерёдно, после каждого запуская убеждаясь, что программа работает корректно и результаты совпадают с полученными вручную.

**Код программы:**

%include "io.inc"

section .text

global CMAIN

CMAIN:

;write your code here

LEA ESI,[a]

;извлекаем значение с индексом 0

MOV AL,[ESI]

MOV DL, 0x3F

AND AL,DL

;PRINT\_UDEC 1,AL

;извлекаем значение с индексом 1

MOV AL,[ESI + 1]

MOV DL, 0x0F

AND AL,DL

SHL AL,2

MOV BL,AL

MOV AL,[ESI]

MOV DL, 0xC0

AND AL,DL

SHR AL,6

OR BL,AL

;PRINT\_UDEC 1,BL

;извлекаем значение с индексом 2

MOV AL,[ESI + 2]

MOV DL, 0x03

AND AL,DL

SHL AL,4

MOV BL,AL

MOV AL,[ESI + 1]

MOV DL, 0xF0

AND AL,DL

SHR AL,4

OR BL,AL

;PRINT\_UDEC 1,BL

;извлекаем значение с индексом 3

MOV AL,[ESI + 2]

MOV DL, 0xFC

AND AL,DL

SHR AL,2

;PRINT\_UDEC 1,AL

;извлекаем значение с индексом 4

MOV AL,[ESI + 3]

MOV DL, 0x3F

AND AL,DL

;PRINT\_UDEC 1,AL

;извлекаем значение с индексом 5

MOV AL,[ESI + 4]

MOV DL, 0x0F

AND AL,DL

SHL AL,2

MOV BL,AL

MOV AL,[ESI + 3]

MOV DL, 0xC0

AND AL,DL

SHR AL,6

OR BL,AL

;PRINT\_UDEC 1,BL

;извлекаем значение с индексом 6

MOV AL,[ESI + 5]

MOV DL, 0x03

AND AL,DL

SHL AL,4

MOV BL,AL

MOV AL,[ESI + 4]

MOV DL, 0xF0

AND AL,DL

SHR AL,4

OR BL,AL

;PRINT\_UDEC 1,BL

;извлекаем значение с индексом 7

MOV AL,[ESI + 5]

MOV DL, 0xFC

AND AL,DL

SHR AL,2

;PRINT\_UDEC 1,AL

ret

section .data

;packed array [15, 21, 7, 12, 59, 30, 25, 11]

a: DB 0x4F, 0x75, 0x30, 0xBB, 0x97, 0x2D

**Задание 3:**

Написать программу, реализующую алгоритм сортировки: сортировка пузырьком **Ход работы**

Сортировка пузырьком подразумевает обмен соседних элементов при каждом проходу алгоритма Например, исходный массив - 4, 1, 2, 9, 3, 5, 8, 6, значит на выходе необходимо получить: 1, 2, 3, 4, 5, 6, 8, 9 Выбрал регистры для хранения входных и выходных значений: EBX – массив Занесла программу в отладчик Выполнив программу и убедившись что результат программы совпадает с вычисленным вручную, занес результаты вычислений в отчет.

Код программы

%include "io.inc"

section .text

global CMAIN

CMAIN:

mov ebp, esp; for correct debugging

LEA EBX, [array]

MOV ECX, 7

next:

PUSH EBX

PUSH ECX

next1:

MOV DL,[EBX]

MOV AL,[EBX+1]

CMP DL, AL

JNL swap

JMP skip

swap:

MOV [EBX+1], DL

MOV [EBX], AL

skip:

ADD EBX,1

LOOP next1

POP ECX

POP EBX

LOOP next

ret

section .data

array: DB 19,4,1,2,9,3,5,0

Вывод: Я научился написанию программ на языке ассемблер с использованием массивов и методов их изменений.