ГУАП КАФЕДРА № 34

ОТЧЕТ

ЗАЩИЩЕН С ОЦЕНКОЙ ПРЕПОДАВАТЕЛЬ

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

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №2 |
| АССЕМБЛЕР |
| по курсу: ИНФОРМАТИКА |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |
| --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 3145 |  | А.А.Одинцов |
|  |  | подпись, дата | инициалы, фамилия |

Санкт-Петербург 2021

# Задание 1

Написать программу, реализующую заданный алгоритм: посчитать количество нечетных элементов массива. Проверить корректность работы программы с помощью отладчика

# Ход работы

Чтобы посчитать количество нечетных массива, необходимо обработать каждый элемент массива и проверить его делимость на 2

Далее суммировать количество нечетных значений и завершить программу

Например, если исходный массив состоит из чисел (4, 5, 3, 2, 1, 8, 2, 2), то количество

значений равна 3

Выбрал регистры для хранения входных и выходных значений: ESI - массив, DL – количество нечетных элементов

Занес программу в отладчик

Выполнил 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:

mov ebp, esp; for correct debugging

;write your code here

XOR EDX,EDX

LEA ESI, [a]

MOV ECX, 8

MOV BL, 2

next:

MOV AL,[ESI]

DIV BL

CMP AH, 0

JNZ summ

next1:

INC ESI

LOOP next

JMP quit

summ:

XOR AH,AH

INC DL

JMP next1

quit:

PRINT\_UDEC 1,DL

ret

section .data

a: DB 4, 5, 3, 2, 1, 8, 2, 2

# Задание 2

Написать программу, реализующую доступ к упакованному массиву с заданной разрядностью элементов: 6 бита

# Ход работы

Выбрал массив, такой, чтобы он состоял из чисел до 64. Например: 32, 47, 5, 18, 61, 50, 24, 12.

Перевел числа в двоичную систему счисления и записал их в строку «слева направо»: 000001, 111101, 101, 01001, 101111, 010011, 00011, 0011

Объединил числа по 64 и перевел в шестнадцатеричную систему счисления: Е0, 5В, 48, BD, 8C, 31

Извлек значения чисел с различными значениями

Выбрал регистры для хранения входных и выходных значений: ESI - массив, AL(BL) – элемент массива

Занес программу в отладчик

Выполнил программу, поочерёдно проверяя каждый элемент, после каждого запуска убеждаясь, что результат программы совпадает с введенными вручную

Занес результаты вычислений в отчет

# 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 [32, 47, 5, 18, 61, 50, 24, 12]

a: DB 0xE0, 0x5B, 0x48, 0xBD, 0x8C, 0x31

# Задание 3

Написать программу, реализующую алгоритм сортировки: сортировка вставками

# Ход работы

Сортировка пузырьком подразумевает что, на каждом шаге алгоритма мы берем один из элементов массива, находим позицию для **вставки** и вставляем.

Например, исходный массив - 8,4,1,2,9,3,5,0, значит на выходе необходимо получить: 0, 1, 2, 3, 4,

5, 8, 9

Выбрал регистры для хранения входных и выходных значений: EBX – массив

Занес программу в отладчик

Выполнил программу и убедился, 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]

lea eax, [array]

MOV ECX, 7

next:

PUSH EBX

PUSH ECX

push eax

l1:

mov dl, [ebx]

l2:

mov dh, [eax-4]

cmp eax, 0

jz skip

cmp dh, dl

jl skip

mov [eax], dh

sub eax, 4

jmp l2

skip:

mov [eax], dl

add ebx, 4

loop l1

pop eax

pop ecx

pop ebx

ret

section .data

array: DD 8,4,1,2,9,3,5,0

**Вывод:** мы познакомились с технологией написания и отладки программ, связанных с массивами, на языке ассемблера, а также базовыми операциями, проводимыми с массивами.