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 Гаврилов Е.М.

подпись, дата инициалы, фамилия

Санкт Петербург 2021.

**Задание 1**

Посчитать количество отрицательных элементов в массиве. Проверить корректность работы программы с помощью отладчика.

**Ход работы**

Для подсчёта количества отрицательных элементов в массиве нужно ввести новую переменную, в которую будем помещать единицу каждый раз, когда при сравнении числа из массива с 0, оно будет меньше.

К примеру, если массив будет – (2; -3; 4; -5;0) – количество отрицательных элементов будет равно 2.

Пусть в EAX хранится количество отрицательных элементов массива, в EBX – адрес массива x.

Я занёс программу в отладчик и выполнил её, убедившись, что результат совпадает с вычислением вручную после каждого шага.

Результат вычислений занёс в ответ.

**Код программы**

CMAIN:

xor eax, eax

mov ecx,5

mov eax,0

lea ebx, [x]

comp:

mov dl, [ebx]

cmp dl, 0

jge next

inc eax

next:

inc ebx

loop comp

ret

section .data

x: DB 2, -3, 4, -5, 0

**Задание 2**

Написать программу, реализующую доступ к упакованному массиву с разрядностью в 3 бита.

**Ход работы**

Задал массив с числами от 0 до 7 включительно. ( 7, 5, 1, 3, 4, 2, 0, 6)

Перевёл его в 2-сс по 3 байта. (111, 101, 001, 011, 100, 010, 000, 110)

Перевернул каждое число. ( 111, 101, 100, 110, 001, 010, 000, 011)

Объединил биты по 8. (1byte 11110110, 2 byte 01100010,3 byte 10000011)

Снова перевернул байты. (1byte 01101111, 2 byte 01000110,3 byte 11000001)

Перевёл каждый байт в 16-сс ( 6f, 46, c1)

Пусть ESI – регистр, хранящий адрес массива X, а EDI – регистр, что будет хранить значения распакованного массива.

Занёс программу в отладчик, проверив правильность вычислений, произведенных вручную на каждом шаге.

**Код программы**

CMAIN:

mov ebp, esp; for correct debugging

;write your code here

xor eax, eax

lea esi,[x]

lea edi,[y]

mov ecx, 8

mov dl,[esi]

mov al,8

next:

cmp al,3

jl ost2

mov ah,dl

and ah,0x7

mov [edi],ah

inc edi

shr dl,3

sub al,3

jmp cycle

ost2: ;здесь мы склеиваем 2 бита старого байта и 1 бит нового, удаляя последний бит нового байта

inc esi

cmp al,2

jl ost1

mov ah,dl

mov dl,[esi]

and dl,0x1

shl dl,2

or ah,dl

mov [edi],ah

inc edi

mov dl,[esi]

shr dl,1

mov al,7

jmp cycle

ost1: ;здесь мы склеиваем 1 бит старого байта и 2 бита нового, удаляя последние 2 бита нового байта

cmp al,1

jl ost0

mov ah,dl

mov dl,[esi]

and dl,0x3

shl dl,1

or ah,dl

mov [edi],ah

inc edi

mov dl,[esi]

shr dl,2

mov al,6

jmp cycle

ost0: ;здесь мы переходим к новому байту, если не осталось бит в старом.

mov dl,[esi]

mov al,8

jmp next

cycle:

loop next

ret

section .data

x: DB 0x6f, 0x46, 0xc1

y: DB 0,0,0,0,0,0,0,0

**Задание 3**

Написать программу, реализующую алгоритм сортировки: шейкерная сортировка.

**Ход работы**

Шейкерная сортировка – сортировка, которая вправо перемещает самое большое число, а влево – самое маленькое, постепенно сокращая границы между элементами.

Пусть нам дан массив: (10, 2, 15, 6, 0, 8, -6), тогда на выходе мы получим: (-6, 0, 2, 6, 8, 10, 15).

Пусть в регистре EAX будет храниться левая граница, а в EBX – правая граница. X – массив.

Занёс программу в отладчик и убедился, что на каждом шаге программы вычисления совпадают с вычислениями вручную.

Занёс результаты вычислений в отчёт.

**Код программы**

CMAIN:

mov ebp, esp ;for correct debugging

xor eax, eax

lea eax, [x]

add eax, 1

lea ebx, [x]

add ebx, 6

top:

cmp eax, ebx

jge exit

mov ecx, ebx

left:

mov esi, ecx

sub esi, 1

mov dl, [esi]

cmp [ecx],dl

jge leftnext

mov dh,[ecx]

mov [esi], dh

mov [ecx], dl

leftnext:

sub ecx, 1

cmp ecx, eax

jge left

add eax, 1

mov ecx, eax

right:

mov esi, ecx

sub esi, 1

mov dl, [esi]

cmp [ecx],dl

jge rightnext

mov dh,[ecx]

mov [esi], dh

mov [ecx], dl

rightnext:

add ecx, 1

cmp ecx, ebx

jle right

sub ebx,1

jmp top

exit:

ret

section .data

x: db 10, 2,15 ,6 ,0 ,8 ,-6

**Вывод: Мы познакомились с построением и отладкой программы на языке Assembler, которые включают в себя массивы и разные операции с ними.**