**МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ**

**УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»**

**РУТ (МИИТ)**

Институт управления и цифровых технологий

Кафедра «Цифровые технологии управления транспортными процессами»

**ОТЧЁТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ № 4**

|  |  |  |
| --- | --- | --- |
| **По**  **дисциплине:** | | Ассемблер |
| **на тему:** | СРЕДСТВА РАЗРАБОТКИ ПРИЛОЖЕНИЙ НА ЯЗЫКЕ АССЕМБЛЕР | |
| **цель работы:** | изучение арифметических команд сложения, вычитания, умножения и деления; - изучение команд сравнения и перехода; - написание и отладка программы вычисления арифметических выражений с ветвлением; | |

|  |  |
| --- | --- |
| **Выполнил студент группы УИС-312** | **(Рыжов В.Р.)** |
| **Принял \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | **(Преподаватель кафедры ЦТУТП Победоносцев И. Н.)** |

**Москва 2022 г.**

**Оглавление**

[ВВЕДЕНИЕ 3](#_Toc122546188)

[Поставленные цели 3](#_Toc122546189)

[Поставленные задачи 3](#_Toc122546190)

[1 БЛОК-СХЕМА 4](#_Toc122546191)

[2 ТЕКСТ ПРОГРАММЫ С КОМЕНТАРИЯМИ 5](#_Toc122546192)

[3 РЕЗУЛЬТАТЫ РАБОТЫ ПРОГРАММЫ 8](#_Toc122546193)

[4 ВОПРОСЫ 9](#_Toc122546194)

[ЗАКЛЮЧЕНИЕ 11](#_Toc122546195)

# ВВЕДЕНИЕ

Массив - структурированный тип данных, состоящий из некоторого числа элементов одного типа.

В Ассемблере массивом можно назвать несколько подряд идущих в памяти байт, слов или двойных слов, но все элементы массива должны быть либо байтами, либо словами, либо двойными словами, т. е. иметь одинаковую длину. В качестве имени массива используется символическое имя адреса (смещения) первого байта первого элемента массива.

## Поставленные цели

В цели работы входит изучение:

* методов косвенной адресации;
* приемов доступа к элементам массивов;
* циклических команд;
* написание и отладка программ обработки одномерных и двумерных массивов.

## Поставленные задачи

1. Написать и отладить программу обработки массивов в соответствии со следующими требованиями. Дан двумерный числовой массив А размерности 4х5, состоящий из положительных и отрицательных чисел. Сформировать одномерный массив В, состоящий из 4–х элементов. Каждый элемент массива В – равен сумме элементов больших своего левого соседа в строчках массива А.

2. Значения исходных данных выбрать самостоятельно, но так, чтобы были задействованы все логические ветви алгоритма программы. Исходные данные и результаты работы программы представляются в памяти.

3. Произвести выполнение программы в отладчике и представить исходную матрицу А и результирующий одномерный массив B в десятичном формате, а также в виде скриншота дампа сегмента данных.

# 1 БЛОК-СХЕМА

Diagram

Description automatically generated

Рисунок 1.1 – блок-схема алгоритма

# 2 ТЕКСТ ПРОГРАММЫ С КОМЕНТАРИЯМИ

;Лабораторная работа №4

;Выполнил ст. гр. УИС-312.

;Вариант 20

;Дан двумерный массив А(4х5).

;Сформировать одномерный массив B(4),каждый элемент которого

;Порядковый номер минимального положительного элемента (или – 1)

;==================================================

.486

.model flat, stdcall

option casemap: none

.stack 100h

;=========================================

include <\masm32\include\kernel32.inc>

includelib <\masm32\lib\kernel32.lib>

;=========================================

.data

N equ 4 ;количество строк

M equ 5 ;количество столбцов

A db 1,2,-3,-4,5

db 6,7,-8,9,0

db -1,-3,-2,-7,-4

db 6,-8,2,0,1

B db 4 dup (?)

m db 255

;==================================================

.code

main:

;для адресации элементов массива A используем регистры

; EBX - смещение 1-го элемента текущей строки (0, 5, 10, 15)

; ESI - смещение элемента внутри текущей строки (0, 1, 2, 3, 4)

;для адресации элементов массива B используем регистр

; EDI - смещение элемента

mov ecx,N ;счетчик строк массива A

;(для организации цикла по строкам)

mov ebx,0 ;смещение первой строки массива A

mov edi,0 ;смещение первого элемента массива B

;-----------------

stroka: ;начало цикла по строкам А (внешний цикл)

mov B[edi],1 ; приравниваем элемень массива к 1

mov edx,ecx ;спасаем ecx: теперь он понадобится

;для организации цикла по столбцам

mov esi,0 ;смещение первого элемента в строке массива A

mov ecx,M ;счетчик столбцов массива A

;(для организации цикла по столбцам)

;-----------------

next: ;начало цикла по столбцам (внутренний цикл)

mov al, A[ebx][esi]; берём текущий элемент массива А

;===================================

cmp al, 0 ; сравниваем текущий элемент массива А с нулем

jl skip; если число отрицательное, то пропускаем его

cmp al, m; сравниваем с минимум

JAE skip; если число больше или равно, то пропускаем его

mov m,al; берём текущей элемент массива как минимальный

;===================================

skip:

add esi,1; переход на следующий элемент строки

loop next ;продолжить, если выбраны не все эл-ты строки

;строки массива A

;-----------------

mov ecx,M; помещаем длину строки

mov esi,0; переход в начало строки

search: ; цикл поиска номера минимального положительного элемента

cmp m,255 ; проверяем был ли элемент массива удв условию

Je exit ; если нет - выходим

mov al, A[ebx][esi]; если да, то ищим его номер

cmp al,m ; текущий элемент равен минимум?

je exit ; если да, то выход

add B[edi],1 ; иначе увеличиваем номер элемента

add esi,1 ; переходим на следующий элемент строки

loop search ; переход к метке search

exit:

mov m,255 ; помещаем в минимум максимальное значение

mov ecx,edx ;восстановим cx

add ebx,M\*type A ;установить bx на начало

;следующей строки массива A(+5)

add edi,type B ;установить di на следующий

;элемент массива B(+1)

add esi,type A ;установить si на следующий

;эл-т в строке(+1)

loop stroka ;продолжить, если проверены не все строки

;-----------------

invoke ExitProcess, 0 ;завершение программы

end main

# 3 РЕЗУЛЬТАТЫ РАБОТЫ ПРОГРАММЫ

На приведенном ниже рисунке 3.1 виден результат работы программы.

* Красный прямоугольник выделяет полученный в ходе работы массив

A picture containing table

Description automatically generated

рисунок 3.1 – результат работы программы

На таблице 3.2-3.3 виден исходный массив А и полученный массив B.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | -3 | -4 | 5 |
| 6 | 7 | -8 | 9 | 0 |
| -1 | -3 | -2 | -7 | -4 |
| 6 | -8 | 2 | 0 | 1 |

рисунок 3.2 – массив А

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 5 | 1 | 4 |

рисунок 3.3 массив B

# 4 ВОПРОСЫ

1. Как описываются одномерные и двумерные массивы в программах на языке ассемблера?

* Перечислением элементов массива в поле операндов одной из директив описания данных.
* Используя оператор повторения dup.
* Использование цикла для инициализации значениями области памяти, которую можно будет впоследствии трактовать как массив.

2. В чем суть метода косвенной адресации? Каким образом формируются косвенные адреса при организации доступа к одномерным и двумерным массивам?

* Косвенная базовая (регистровая) адресация. При такой адресации эффективный адрес операнда может находиться в любом из регистров общего назначения, кроме sp/esp и bp/ebp (это специфические регистры для работы с сегментом стека). Синтаксически в команде этот режим адресации выражается заключением имени регистра в квадратные скобки [].

mov eax, [esi] ; eax = \*esi; \*esi значение по адресу esi

Данный способ адресации позволяет динамически назначить адрес операнда для некоторой машинной команды и применяется при организации циклических вычислений и при работе со структурами данных, массивами.

3. Элемент массива MAS(10x20) адресуется с помощью выражения MAS[EBX][ESI]. Какие значения регистров EBX и ESI следует установить при обращении к элементу массива MAS, размещенному в 3-й строке и 5-м столбце, если исходный массив а) имеет тип DB; б) имеет тип DW; в) имеет тип DD?

4. Поясните выражение ($-MAS)/TYPE MAS, представленное в примере п. 5.4.

5. Каковы особенности используемых в языке ассемблера циклических команд. Можно ли использовать в качестве счетчика цикла вместо регистра ECX другой регистр?

* Команда цикла в Ассемблере всегда уменьшает значение счётчика на единицу. Это значение находится в регистре СХ (или ECX). Отличия между командами цикла заключаются только в условиях, при которых выполняется переход к метке или цикл завершается.

6. Сколько повторений обеспечит команда LOOP, если начальное значение ECX=0?

* Алгоритм работы команды LOOP:
  + CX = CX - 1
  + Если CX не равен 0, то выполнить переход (продолжить цикл)
  + Иначе не выполнять переход (прервать цикл и продолжить выполнение программы)

7. Познакомьтесь с приведенным в п.5.8 примером программы обработки двумерного массива. Объясните: - какие значения принимают регистры EBX, ESI, EDI, ECX в ходе работы программы; - что означает и для чего используется выражение M\*TYPE A; - какое значение будет в регистре AL после выполнения команды MOV AL,A[EBX][ESI] если а) EBX=0,ESI=4; б) EBX=5,ESI=2; в) EBX=10,ESI=3; г) EBX=1,ESI=1; д) EBX=8,ESI=0 ?

# ЗАКЛЮЧЕНИЕ

В ходе данной работы я научился работать с массивами, методами альтернативной адресации в коде для ассемблера и циклическими командами, для дальнейшего их использования, при написании кода. А также лучше понял алгоритмы работы программ, написанных для ассемблера.