Учреждение образования Республики Беларусь

«Гомельский государственный технический университет им. П. О. Сухого»

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

Отчет по лабораторной работе №4

по дисциплине «Архитектура ЭВМ»

по теме «Типы данных Ассемблера. Пересылка данных»

Выполнил:

студент группы ИТП-21

Половцев М. С.

Проверил:

преподаватель

Савельев В. А.

Гомель, 2022

**Цель:** знакомство с типами переменных Ассемблера и их размещением в памяти, изучение команд пересылки данных Ассемблера, получение навыков работы с отладчиком.

**Ход работы**

Задание

1. Написать программу, где в сегменте данных будут созданы следующие переменные:

Des = 65;

Nex = -160;

Bin1 = 0b1000111011;

Bin2 = 0b1011110000;

Hex = 0xAD456C4;

Text = ‘Good bye, America’;

Array[5] = {256, 765, 89, 654, 9}.

Для создания переменных выделить минимально необходимый объем памяти.

В тексте программы выполнить очистку регистров EAX и EBX с помощью команд MOV и XOR соответственно.

Переместить переменную Bin1 в регистр EAX и логически умножить ее на переменную Bin2. Результат сохранить в переменную Bin3.

Сохранить переменные Des и Hex в стек. Извлечь переменные из стека, поменяв местами их значения.

Листинг:

.386

.model flat, stdcall

option casemap:none

include C:\masm32\include\windows.inc

include C:\masm32\include\kernel32.inc

includelib C:\masm32\lib\kernel32.lib

; Сегмент данных

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

.data

Des dd 65

Nep dw -160

Bin1 dd 1000111011b

Bin2 dd 1011110000b

Hex dd 0AD456C4h

Text db "Good bye, America", 0

Array dw 256, 765, 89, 654, 9

.data?

Bin3 dd ?

; Сегмент кода

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

.code

start:

; Очистка регистров eax и ebx

mov eax, 0

xor ebx, ebx

; Сохранение переменной Bin1 в регистр eax и умножение её на переменную Bin2

mov eax, Bin1

and eax, Bin2

mov Bin3, eax

; Сохранение переменных Des и Hex в стек и замена значений переменных

push Des

push Hex

pop Des

pop Hex

; задержка

push INFINITE

call Sleep

; завершаем процесс

push NULL

call ExitProcess

end start

2. Открыть программу в отладчике. Указать адреса расположения переменных в памяти, заполнив таблицу:

|  |  |  |
| --- | --- | --- |
| Имя переменной | Адрес | Порядок байт в памяти |
| Des | 00403000 | 4100 |
| Nex | 00403004 | 60FF |
| Bin1 | 00403006 | 3B020000 |
| Bin2 | 0040300A | F0020000 |
| Hex | 0040300E | C456D40A |
| Text | 00403012 | 476F6F64206279652C20416D6572696361 |
| Array | 00403023 | 0A000001FD02598E02 |

3. Выполнить программу в пошаговом режиме. После выполнения каждого шага заносить данные в таблицу

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| EAX | EBX | Bin3 | ESP | FC | FZ | FP | FS |
| 0019FFCC | 002C1000 | 00000000 | 0019FF74 | 0 | 1 | 1 | 0 |
| 00000000 | 002C1000 | 00000000 | 0019FF74 | 0 | 1 | 1 | 0 |
| 00000000 | 00000000 | 00000000 | 0019FF74 | 0 | 1 | 1 | 0 |
| 0000023B | 00000000 | 00000000 | 0019FF74 | 0 | 1 | 1 | 0 |
| 00000230 | 00000000 | 00000000 | 0019FF74 | 0 | 0 | 1 | 0 |
| 00000230 | 00000000 | 00000023 | 0019FF70 | 0 | 0 | 1 | 0 |
| 00000230 | 00000000 | 00000023 | 0019FF6C | 0 | 0 | 1 | 0 |
| 00000230 | 00000000 | 00000023 | 0019FF70 | 0 | 0 | 1 | 0 |
| 00000230 | 00000000 | 00000023 | 0019FF74 | 0 | 0 | 1 | 0 |
| 00000230 | 00000000 | 00000023 | 0019FF70 | 0 | 0 | 1 | 0 |

4. Выводы по пункту 2: размещение инициализированных данных начинается с адреса 00403000. Каждое значение занимает столько байт, сколько резервируется соответствующей директивой. В отладчике каждый байт представлен двумя шестнадцатеричными цифрами. Используется обратный порядок байт, т.е. младший байт числа находится в младших адресах памяти (перед старшим). Если шестнадцатеричная комбинация соответствует коду символа, то он высвечивается в следующем столбце (ASCII), иначе в нем высвечивается точка.

Выводы по пункту 3: адрес начала программы 00401000. При каждом выполнении команды мы можем наблюдать изменение данных в памяти и/или в регистрах, отслеживая процесс выполнения программы и контролируя правильность промежуточных результатов.

1 шаг: очистка регистра EAX.

2 шаг: очистка регистра EBX.

3 шаг: помещение в регистр значения 0000023B.

4 шаг: выполнение логического умножения и занесение результата в регистр EAX.

5 шаг: копирование результата в переменную Bin3 из регистра EAX.

6 шаг: занесение значение переменной Des в стек.

7 шаг: занесение значение переменной Hex в стек.

8 шаг: извлечение значения переменной Hex из стека.

9 шаг: извлечение значения переменной Des из стека.

**Вывод:** к основным директивам определения данных относятся db (1 байт), dw (2 байта), dd (4 байта) и т. д. Для пересылки данных используется команда Ассемблера MOV. Она позволяет копировать содержимое источника и помещать это содержимое в приёмник. Средствами отладчика можно пошагово отследить выполнение программы (например, содержимое регистров).