**Министерство науки и высшего образования Российской Федерации**

**федеральное государственное автономное образовательное учреждение высшего образования**

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**УНИВЕРСИТЕТ ИТМО»**

**ФАКУЛЬТЕТ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**ОТЧЕТ**

**по Лабораторной работе № 3**

**«Управление памятью с использованием сегментов»**

Специальность 09.02.07«Информационные системы и программирование»

Дисциплина «Архитектура аппаратных средств»

|  |  |
| --- | --- |
| Преподаватель:  Ошурок Э.Э.  Ассистент:  Теричев М.С.  «\_\_» \_\_\_\_\_\_ 2020 г.  Оценка \_\_\_\_\_\_\_\_\_ | Выполнил:  студент группы Y2337  Князев А. А. |

Санкт-Петербург

2019/2020

# ЗАДАНИЕ

Цель работы – изучение способов организации стека и множеств данных в памяти, исследование порядка функционирования ЭВМ при выполнении таких программ.

Задание: написать многосегментную программу, в которой реализовать процедуру копирования каждого второго числового значения из произвольного числового ряда в отдельный сегмент данных

Таблица 1. Текст исходной программы

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес | Код команды | Мнемоника | Комментарии |
|  |  | org 0x100 |  |
| 0100 | B90400 | mov cx, 4 | Размер массива |
| 0103 | E682E01 | Push 012F |  |
| 0106 | E80100 | Call 010A |  |
| 0109 | C3 | ret |  |
|  |  | proc: |  |
| 010A | 55 | push bp |  |
| 010B | 89E5 | mov bp,sp | Теперь bp указывает на вершину стека |
| 010D | 8B7604 | mov si, [bp+4] |  |
| 0110 | 83C604 | Add si, 2 | Переход на второй элемент массива |
| 0113 | 8CDB | Mov bx, ds | Запоминаем адрес первого сегмента |
| 0115 | 8CDA | Mov dx, ds |  |
|  |  | while: |  |
| 0117 | 8B04 | mov ax, [si] | В ах перемещаем элемент из массива |
| 0119 | 83C604 | Add si,4 | Сдвигаемся на два элемента в массиве |
| 011C | 81C20010 | Add dx, 0x1000 | Переходим на следующий сегмент |
| 0120 | 8EDA | Mov ds, dx | Записываем элемент из массива |
| 0122 | BF2000 | Mov di, 0x0020 |
| 0125 | 8905 | Mov [di], ax |
| 0127 | 8EDB | Mov ds,bx | Возвращаемся на первый сегмент |
| 0129 | 83E902 | Sub cx,2 |  |
| 012C | 75E9 | Jnz while |  |
| 012E | C3 | ret |  |
| 128 | 75ED | Array dw 1,2,3,4 |  |

Таблица 2. Трассировка

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Адрес | Команда | AX | BX | CX | DX | IP | OF | DF | IF | SF | ZF | AF | PF | CF | Память | Стек | Аргументы команды |
| 0100 | mov | 0000 | 0000 | 0004 | 0000 | 0103 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |  | 0000 | CX,0004 |
| 0103 | push | 0000 | 0000 | 0004 | 0000 | 0106 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |  | 012F | 012F |
| 0106 | call | 0000 | 0000 | 0004 | 0000 | 010A | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |  | 0109 | 010A |
| 010A | push | 0000 | 0000 | 0004 | 0000 | 010B | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |  | 0000 | BP |
| 010B | mov | 0000 | 0000 | 0004 | 0000 | 010D | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |  | 0000 | BP,SP |
| 010D | mov | 0000 | 0000 | 0004 | 0000 | 0110 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | BP+04 | 0000 | SI,[BP+04] |
| 0110 | add | 0000 | 0000 | 0004 | 0000 | 0113 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |  | 0000 | SI,0002 |
| 0113 | mov | 0000 | 19F5 | 0004 | 0000 | 0115 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |  | 0000 | BX,DS |
| 0115 | mov | 0000 | 19F5 | 0004 | 19F5 | 0117 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |  | 0000 | DX,DS |
| 0117 | mov | 0002 | 19F5 | 0004 | 19F5 | 0119 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0131 | 0000 | AX,[SI] |
| 0119 | add | 0002 | 19F5 | 0004 | 19F5 | 011C | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |  | 0000 | SI,0004 |
| 011C | add | 0002 | 19F5 | 0004 | 29F5 | 0120 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |  | 0000 | DX,1000 |
| 0120 | mov | 0002 | 19F5 | 0004 | 29F5 | 0122 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |  | 0000 | DS,DX |
| 0122 | mov | 0002 | 19F5 | 0004 | 29F5 | 0125 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |  | 0000 | DI,0020 |
| 0125 | mov | 0002 | 19F5 | 0004 | 29F5 | 0127 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0020 | 0000 | [DI],AX |
| 0127 | mov | 0002 | 19F5 | 0004 | 29F5 | 0129 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |  | 0000 | DS,BX |
| 0129 | sub | 0002 | 19F5 | 0002 | 29F5 | 012C | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |  | 0000 | CX,0002 |
| 012C | jnz | 0002 | 19F5 | 0002 | 29F5 | 0117 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |  | 0000 | 0117 |
| 0117 | mov | 0004 | 19F5 | 0002 | 29F5 | 0119 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0135 | 0000 | AX,[SI] |
| 0119 | add | 0004 | 19F5 | 0002 | 29F5 | 011C | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |  | 0000 | SI,0004 |
| 011C | add | 0004 | 19F5 | 0002 | 39F5 | 0120 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |  | 0000 | DX,1000 |
| 0120 | mov | 0004 | 19F5 | 0002 | 39F5 | 0122 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |  | 0000 | DS,DX |
| 0122 | mov | 0004 | 19F5 | 0002 | 39F5 | 0125 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |  | 0000 | DI,0020 |
| 0125 | mov | 0004 | 19F5 | 0002 | 39F5 | 0127 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0020 | 0000 | [DI],AX |
| 0127 | mov | 0004 | 19F5 | 0002 | 39F5 | 0129 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |  | 0000 | DS,BX |
| 0129 | sub | 0004 | 19F5 | 0000 | 39F5 | 012C | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |  | 0000 | CX,0002 |
| 012C | jnz | 0004 | 19F5 | 0000 | 39F5 | 0000 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |  | 0000 | 0117 |
| 012E | ret |  |  |  |  |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |  |  |

3. Листинг программы:

org 0x100

mov cx,4

push Array

call func

ret

func:

push bp

mov bp,sp

mov si,[bp+4]

add si,2

mov bx,ds

mov dx, ds

while:

mov ax,[si]

add si,4

add dx, 0x1000

mov ds, dx

mov di, 0x0020

mov [di], ax

mov ds,bx

sub cx,2

jnz while

ret

Array dw 1,2,3,4