МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ

ХАРЬКОВСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ РАДИОЭЛЕКТРОНИКИ

Кафедра СТ

Отчет

по лабораторной работе № 1

по дисциплине: «Программирование(системное)»

Выполнил: Проверил:

ст. гр. АКТСИу-17-1 Коваленко А. И.

Черкашин В.А.

Харьков 2018

1 Исследование триггеров

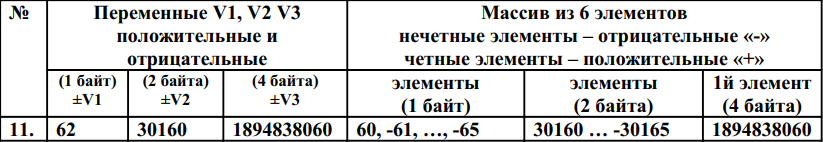
* 1. Цель работы

Целью работы является изучение универсального двоичного счётчика и приобретение навыков в построении и экспериментальном исследовании счётчиков.

* 1. Задания

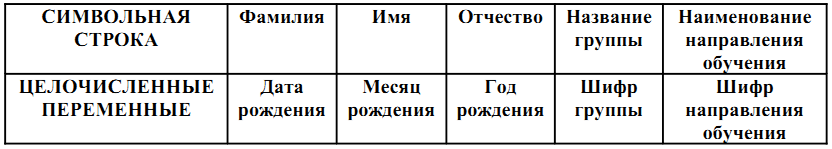
Задание 1. В соответствии с заданным вариантом табл. 1 описать положительные и отрицательные целочисленные данные на языке Ассемблера. Создать исполняемый файл (\*.exe) и проконтролировать создаваемый сегмент данных с помощью отладчика OllyDbg.

Таблица 1 – Вариант задания



Задание 2. Описать данные на языке Ассемблера в соответствии с табл. 2: Создать исполняемый файл (\*.exe) и проконтролировать создаваемый сегмент данных с помощью отладчика OllyDbg. С использованием API функции MessageBox вывести описанные данные в окнах сообщений. В сообщениях использовать различные число и виды кнопок, а также все 4-ре вида пиктограмм. Продублировать один из вариантов вывода числовых данных, используя шестнадцатеричную систему счисления.

Таблица 2 – Описываемые данные и их тип



Задание 3. Создать программу на языке Ассемблера, осуществляющую консольный ввод-вывод. Программа должна обеспечивать ввод данных (описанных в табл. 2) с консоли и вывод их на консоль.

* 1. Ход работы

1.3.1 Листинг к первому заданию.

;Задание №1 Вариант 11 Черкашин В.А. АКТСИу 17-2

.486 ;используется набор команд i80486

.model flat, stdcall ;объявление используемой модели памяти

option casemap: none ;опция включает различимость прописных и строчных букв

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

.data ;объявление сегмента данных

DB 62 ;объявление 1-байтного целого без имени

DB 62d ;используем директиву DB

BYTE 111110b ;используем директиву BYTE

BYTE 3Eh ;1-байтныое число в 2,8,16,10-ном коде

BYTE 76q ;62d = 111110b = 3Eh = 76q

SBYTE 62d ;используем директиву SBYTE

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

DW 30160 ;объявление 2-байтного целого без имени

DW 30160d ;используем директиву DW

WORD 111010111010000b ;используем директиву WORD

WORD 75D0h ;2-байтныое число в 2,8,16,10-ном коде

WORD 72720q ;30160d = 111010111010000b = 75D0h = 72720q

SWORD 30160d ;используем директиву SWORD

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

DD 1894838060 ;объявление 4-байтного целого без имени

DD 1894838060d ;используем директиву DD

DWORD 1110000111100001110111100101100b ;используем директиву DWORD

DWORD 70F0EF2Ch ;4-байтныое число в 2,8,16,10-ном коде

DWORD 16074167454q ;1894838060d = 1110000111100001110111100101100b = 70F0EF2Ch = 16074167454q

SDWORD 1894838060d ;используем директиву SDWORD

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

DB -62 ;объявление 1-байтного знакового без имени

DB -62d ;используем директиву DB

SBYTE 11000010b ;используем директиву SBYTE

SBYTE 0C2h ;1-байтныое знаковое число в 2,8,16,10-ном коде

SBYTE 302q ;-62d = 11000010b = 0C2h = 302q

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

DW -30160 ;объявление 2-байтного числа со знаком без имени

DW -30160d ;используем директиву DW

SWORD 1000101000110000b ;используем директиву SWORD

SWORD 8A30h ;2-байтныое знаковое число в 2,8,16,10-ном коде

SWORD 105060q ;30160d = 111010111010000b = 75D0h = 72720q

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

DD -1894838060 ;объявление 4-байтного знакового без имени

DD -1894838060d ;используем директиву DD

SDWORD 10001111000011110001000011010100b ;используем директиву SDWORD

SDWORD 8F0F10D4h ;4-байтныое знаковое число в 2,8,16,10-ном коде

SDWORD 21703610324q ;-1894838060d = 10001111000011110001000011010100b = 8F0F10D4h = 21703610324q

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

DB 60, -61, 62, -63, 64, -65

SBYTE 60d, -61d, 62d, -63d, 64d, -65d

DB 111100b, 11000011b, 11000001b, 111111b, 1000000b, 10111111b

SBYTE 3Ch, 0C3h, 0C1h, 3Fh, 40h, 0BFh

DB 74q, 303q, 301q, 77q, 100q, 277q

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

DW 30160, -30161, 30162, -30163, 30164, -30165

SWORD 30160d, -30161d, 30162d, -30163d, 30164d, -30165d

DW 111010111010000b, 1000101000101111b, 111010111010010b, 1000101000101101b, 111010111010100b, 1000101000101011b

SWORD 75D0h, 8A2Fh, 75D2h, 8A2Dh, 75D4h,

DW 72720q, 105057q, 72722q , 105055q , 72724q, 105053q

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

DD 1894838060, -1894838061, 1894838062, -1894838063, 1894838064, -1894838065

SDWORD 1894838060d, -1894838061d, 1894838062d, -1894838063d, 1894838064d, -1894838065d

DD 1110000111100001110111100101100b,

10001111000011110001000011010011b,

1110000111100001110111100101110b,

10001111000011110001000011010001b,

1110000111100001110111100110000b,

10001111000011110001000011001111b

SDWORD 70F0EF2Ch, 8F0F10D3h, 70F0EF2Eh, 8F0F10D1h, 70F0EF30h, 8F0F10CFh

DD 16074167454q , 21703610323q, 16074167456q, 21703610321q, 16074167460q, 21703610317q

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

.code ;объявление сегмента кода

start: ;start - метка начала основной программы

ret ;возврат управления ОС Windows

end start ;объявление конца основной программы

Переменный описаны без имени, так как мы их нигде не используем. На рис.1.1 можно посмотреть дамп памяти, где видно, как данные расположились в памяти.

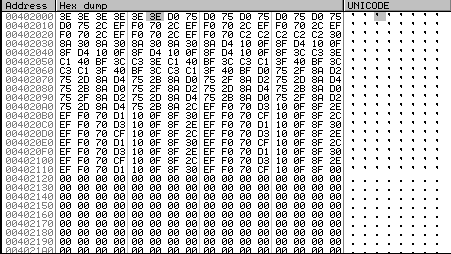


Рис 1.1 Дамп памяти для первого задания

1.3.2 Листинг задания №2

;Задание №2 Вариант 11 Черкашин В.А. АКТСИу 17-2

.486 ;используется набор команд i80386

.model flat, stdcall ;используемая модель памяти FLAT

option casemap: none

include \masm32\include\windows.inc ;подключаемые библиотеки

include \masm32\include\user32.inc

include \masm32\include\kernel32.inc

includelib \masm32\lib\user32.lib

includelib \masm32\lib\kernel32.lib

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

.data ;сегмент данных

MsgBoxTitle db "Вывод информации",0 ;строка заголовка

MsgBoxText db 128 dup(?) ;буфер вывода

FirstName db "Владимир",0 ;инициализация имени

LastName db "Черкашин",0 ;инициализация фамилии

MiddleName db "Андреевич" ,0 ;инициализация отчества

NameOfGroup db "АКТСИу 17-2" ,0 ;инициализация имени группы

DirectionOfLearn db "АКТАКИТ",0 ;инициализация направления обучения

Day db 27 ;инициализация дня рождения

Mounth db 9 ;инициализация месяца рождения

Year dw 1997 ;инициализация года рождения

GroupCode db 2 ;Инициализация шифра группы

LearnCode db 151 ;инициализция направления обучения

format1 db "Дата рождения студента", ;строка спецификации формата

0dh,0ah, "День %d", 0dh,0ah, "Месяц %d", 0dh,0ah, "Год %d",0

format2 db "АКТСИУ 17-%d",0

format3 db "АКТАКИТ",0dh,0ah, "Шифр %d",0

format4 db "Дата рождения студента H", ;строка спецификации формата

0dh,0ah, "День %0X", 0dh,0ah, "Месяц %0X", 0dh,0ah, "Год %0X",0

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

.code ;сегмент кода

start:

invoke MessageBox, 0, ;invoke – вызов функции API MessageBox

ADDR FirstName, ;ADDR нахождение адреса FirstName

ADDR MsgBoxTitle, ;ADDR нахождение адреса MsgBoxTitle

MB\_OK OR MB\_ICONEXCLAMATION ;кнопка «OK»

invoke MessageBox, 0, ;invoke – вызов функции API MessageBox

ADDR LastName,

;ADDR нахождение адреса LastName

ADDR MsgBoxTitle, ;ADDR нахождение адреса MsgBoxTitle

MB\_YESNO OR MB\_ICONINFORMATION OR MB\_DEFBUTTON1 ;кнопки

invoke MessageBox, 0, ;invoke – вызов функции API MessageBox

ADDR MiddleName, ;ADDR нахождение адреса MiddleName

ADDR MsgBoxTitle, ;ADDR нахождение адреса MsgBoxTitle

MB\_OKCANCEL OR MB\_ICONQUESTION OR MB\_DEFBUTTON2 ;кнопки

invoke MessageBox, 0, ;invoke – вызов функции API MessageBox

ADDR NameOfGroup, ;ADDR нахождение адреса NameOfGroup

ADDR MsgBoxTitle, ;ADDR нахождение адреса MsgBoxTitle

MB\_YESNOCANCEL OR MB\_ICONSTOP OR MB\_DEFBUTTON3 ;кнопки

invoke MessageBox, 0, ;invoke – вызов функции API MessageBox

ADDR DirectionOfLearn, ;ADDR нахождение адреса DirectionOfLearn

ADDR MsgBoxTitle, ;ADDR нахождение адреса MsgBoxTitle

MB\_RETRYCANCEL OR MB\_ICONQUESTION OR MB\_DEFBUTTON2 ;кнопки

movzx eax,Day

push eax

movzx eax,Mounth

push eax

movzx eax,Year

push eax

LEA eax, format1

push eax

LEA eax, MsgBoxText

push eax

call wsprintf

add esp,5\*4

invoke MessageBox, 0, ;invoke – вызов функции API MessageBox

ADDR MsgBoxText, ;ADDR нахождение адреса DirectionOfLearn

ADDR MsgBoxTitle, ;ADDR нахождение адреса MsgBoxTitle

MB\_OKCANCEL OR MB\_YESNO OR MB\_DEFBUTTON1 OR MB\_ICONSTOP ;кнопки

invoke wsprintf,addr MsgBoxText,addr format2,GroupCode

invoke MessageBox, 0, ;invoke – вызов функции API MessageBox

ADDR MsgBoxText, ;ADDR нахождение адреса DirectionOfLearn

ADDR MsgBoxTitle, ;ADDR нахождение адреса MsgBoxTitle

MB\_YESNO OR MB\_DEFBUTTON1 OR MB\_ICONSTOP ;кнопки

invoke wsprintf,addr MsgBoxText,addr format3,LearnCode

invoke MessageBox, 0, ;invoke – вызов функции API MessageBox

ADDR MsgBoxText, ;ADDR нахождение адреса DirectionOfLearn

ADDR MsgBoxTitle, ;ADDR нахождение адреса MsgBoxTitle

MB\_YESNO OR MB\_DEFBUTTON1 OR MB\_ICONSTOP ;кнопки

movzx eax,Day

movzx edx,Mounth

movzx ecx,Year

invoke wsprintf,addr MsgBoxText,addr format4,eax,edx,ecx

invoke MessageBox, 0, ;invoke – вызов функции API MessageBox

ADDR MsgBoxText, ;ADDR нахождение адреса DirectionOfLearn

ADDR MsgBoxTitle, ;ADDR нахождение адреса MsgBoxTitle

MB\_OKCANCEL OR MB\_YESNO OR MB\_DEFBUTTON1 OR MB\_ICONSTOP ;кнопки

invoke ExitProcess, 0 ;invoke – вызов функции API

end start

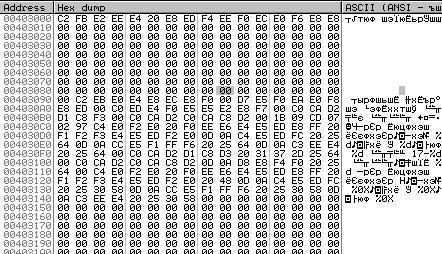


Рис 1.2 Дамп памяти ко второму заданию.

Во втором задание функция wsprintf в masm 32 забагованная и возможно не использует соглашения о вызовах, в листинге были показаны два способа обхода этой проблемы – это вызов через call и стек, или вызовом invoke, но перед этим данные передаются в регистры.

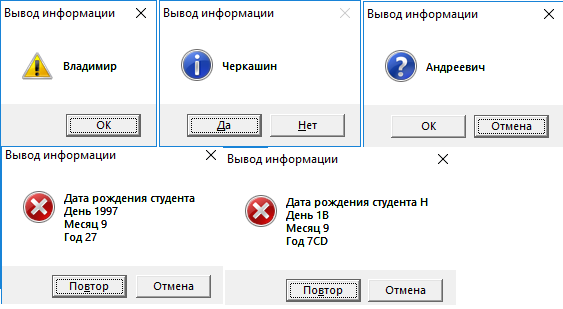


Рис 1.3 Выводимые окна программой во втором задании

1.3.3) Листинг к зданию 3

;Черкашин В.А. АКТСИу17-2 лаб 1 Задание 3

.486 ;используется набор команд i80486

.model flat, stdcall ;используемая модель памяти FLAT

option casemap: none

include \masm32\include\windows.inc

include \masm32\include\user32.inc

include \masm32\include\kernel32.inc

include \masm32\include\msvcrt.inc ;для crt\_atoi

includelib \masm32\lib\user32.lib

includelib \masm32\lib\kernel32.lib

includelib \masm32\lib\msvcrt.lib ;для crt\_atoi

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

.data ;сегмент данных

titleStr db "Консоль: вывод и ввод данных",0 ;заголовок консоли

MsgBoxTitle db "Вывод информации",0 ;строка заголовка

MsgBoxText db "Консоль завершила работу",0 ;строка вывода

FirstName db 15 dup(?) ;инициализация имени

LastName db 16 dup(?) ;инициализация фамилии

MiddleName db 16 dup(?) ;инициализация отчества

NameOfGroup db 16 dup(?) ;инициализация имени группы

DirectionOfLearn db 16 dup(?) ;инициализация направления обучения

Day db ? ;инициализация дня рождения

Mounth db ? ;инициализация месяца рождения

Year dw ? ;инициализация года рождения

GroupCode db ? ;Инициализация шифра группы

LearnCode db ? ;инициализция направления обучения

buf1 BYTE "Введение данных",

0,0Ah,0Dh ;буфер1 для выводимой строки + перевод строки

buf2 BYTE "Введите Фамилию",0 ;буфер2 для выводимой строки

buf3 BYTE "Введите Имя",0 ;буфер3 для выводимой строки

buf4 BYTE "Введите Отчество",0 ;буфер4 для выводимой строки

buf5 BYTE "Введите Название группы",0 ;буфер5 для выводимой строки

buf6 BYTE "Введите Направление обучения",0 ;буфер6 для выводимой строки

buf7 BYTE "Введите День рождения",0 ;буфер7 для выводимой строки

buf8 BYTE "Введите Месяц рождения (числом)",0 ;буфер8 для выводимой строки

buf9 BYTE "Введите Год рождения",0 ;буфер9 для выводимой строки

buf10 BYTE "Введите Шифр группы",0 ;буфер10 для выводимой строки

buf11 BYTE "Введите Шифр направления обучения",0 ;буфер11 для выводимой строки

outHandle DWORD ? ;ячейка памяти для хэндла вывода

inHandle DWORD ? ;ячейка памяти для хэндла ввода

namberW DWORD ? ;количество выводимых символов

namberR DWORD ? ;количество вводимых символов

format1 db 0dh, "Вывод на консоль: Фамилия студента: %hS Имя студента: %hS Отчество: %hS Группа: %hS Направления обучения: %hS",0

format2 db 0dh, "День: %d Месяц: %d Год: %d Шифр группы: %d Шифр направления обучения: %d",0

;===================================буферы===========================================

NumberBuf db 30 dup (?) ;буффер для чисел в 30 байт

TextBuf db 255 dup (?) ;буффер для строк в 255 байт

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

.code ;сегмент кода

start:

invoke AllocConsole ;запрашиваем у ОС Windows консоль

;перекодируем titleStr как источник и получатель из Win1251 -> DOS

invoke CharToOem, ;invoke - вызов функции API CharToOem

ADDR titleStr, ;ADDR- определение адреса titleStr получателя

ADDR titleStr ;ADDR- определение адреса titleStr источника

invoke SetConsoleTitle, ;invoke - вызов функции API SetConsoleTitle

ADDR titleStr ; ;ADDR- определение адреса titleStr

invoke GetStdHandle, ;invoke - вызов функции API GetStdHandle

STD\_INPUT\_HANDLE ;получаем хэндл ввода

MOV inHandle, EAX ;сохраняе хэндл вывода в ячейке inHandle

invoke GetStdHandle, ;invoke - вызов функции API GetStdHandle

STD\_OUTPUT\_HANDLE ;получаем хэндл вывода

MOV outHandle, EAX ;сохраняем хэндл вывода в ячейке outHandle

invoke CharToOem, ADDR buf1, ADDR buf1

invoke CharToOem, ADDR buf2, ADDR buf2

invoke CharToOem, ADDR buf3, ADDR buf3

invoke CharToOem, ADDR buf4, ADDR buf4

invoke CharToOem, ADDR buf5, ADDR buf5

invoke CharToOem, ADDR buf6, ADDR buf6

invoke CharToOem, ADDR buf7, ADDR buf7

invoke CharToOem, ADDR buf8, ADDR buf8

invoke CharToOem, ADDR buf9, ADDR buf9

invoke CharToOem, ADDR buf10,ADDR buf10

invoke CharToOem, ADDR buf11, ADDR buf11

invoke WriteConsoleA, outHandle, ADDR buf1, SIZEOF buf1, ADDR namberW, NULL

invoke WriteConsoleA, outHandle, ADDR buf2, SIZEOF buf2, ADDR namberW, NULL

invoke ReadConsole, inHandle, ADDR FirstName, SIZEOF FirstName, ADDR namberR, NULL

invoke OemToChar,ADDR FirstName,ADDR FirstName

invoke WriteConsoleA, outHandle, ADDR buf3, SIZEOF buf3, ADDR namberW, NULL

invoke ReadConsole, inHandle, ADDR LastName, SIZEOF LastName, ADDR namberR, NULL

invoke OemToChar,ADDR LastName,ADDR LastName

invoke WriteConsoleA, outHandle, ADDR buf4, SIZEOF buf4, ADDR namberW, NULL

invoke ReadConsole, inHandle, ADDR MiddleName, SIZEOF MiddleName,ADDR namberR, NULL

invoke OemToChar,ADDR MiddleName,ADDR MiddleName

invoke WriteConsoleA, outHandle, ADDR buf5, SIZEOF buf5, ADDR namberW, NULL

invoke ReadConsole, inHandle, ADDR NameOfGroup, SIZEOF NameOfGroup,ADDR namberR, NULL invoke OemToChar,ADDR NameOfGroup,ADDR NameOfGroup

invoke WriteConsoleA, outHandle, ADDR buf6, SIZEOF buf6, ADDR namberW, NULL

invoke ReadConsole, inHandle,ADDR DirectionOfLearn,SIZEOF DirectionOfLearn,ADDR namberR, NULL invoke OemToChar,ADDR DirectionOfLearn,ADDR DirectionOfLearn

invoke WriteConsoleA, outHandle, ADDR buf7, SIZEOF buf7, ADDR namberW, NULL

invoke ReadConsole, inHandle, ADDR NumberBuf,SIZEOF NumberBuf, ADDR namberR, NULL

invoke crt\_atoi, addr NumberBuf

MOV Day, AL

invoke WriteConsoleA, outHandle, ADDR buf8, SIZEOF buf8, ADDR namberW, NULL

invoke ReadConsole, inHandle, ADDR NumberBuf,SIZEOF NumberBuf,ADDR namberR, NULL

invoke crt\_atoi, addr NumberBuf

MOV Mounth, AL

invoke WriteConsoleA, outHandle, ADDR buf9, SIZEOF buf9, ADDR namberW, NULL

invoke ReadConsole, inHandle, ADDR NumberBuf, SIZEOF NumberBuf, ADDR namberR, NULL

invoke crt\_atoi, addr NumberBuf

MOV Year, AX

invoke WriteConsoleA, outHandle, ADDR buf10, SIZEOF buf10, ADDR namberW, NULL

invoke ReadConsole, inHandle, ADDR NumberBuf,SIZEOF NumberBuf, ADDR namberR, NULL

invoke crt\_atoi, addr NumberBuf

MOV GroupCode, AL

invoke WriteConsoleA, outHandle, ADDR buf11, SIZEOF buf11, ADDR namberW, NULL

invoke ReadConsole, inHandle, ADDR NumberBuf,SIZEOF NumberBuf,ADDR namberR, NULL

invoke crt\_atoi, addr NumberBuf

MOV LearnCode, AL

LEA eax,DirectionOfLearn

push eax

LEA eax, NameOfGroup

push eax

LEA eax, MiddleName

push eax

LEA eax, LastName

push eax

LEA eax, FirstName

push eax

LEA eax, format1

push eax

LEA eax, TextBuf

push eax

call wsprintf

add esp,7\*4

invoke CharToOem, ADDR TextBuf, ADDR TextBuf

invoke WriteConsoleA, outHandle, ADDR TextBuf, SIZEOF TextBuf, ADDR namberW, NULL

cld ;снять флаг DF

lea di, TextBuf

mov cx, SIZEOF TextBuf

sub al, al ;обнуляем записываемый байт

rep stosb ;обнуляем массив байтов

movzx eax,LearnCode

push eax

movzx eax,GroupCode

push eax

movzx eax,Year

push eax

movzx eax, Mounth

push eax

movzx eax, Day

push eax

LEA eax, format2

push eax

LEA eax, TextBuf

push eax

call wsprintf

add esp,7\*4

invoke CharToOem, ADDR TextBuf, ADDR TextBuf

invoke WriteConsoleA, outHandle, ADDR TextBuf, SIZEOF TextBuf, ADDR namberW, NULL

;чтобы консоль не закрылась, вызываем MessageBox

invoke MessageBox, 0, ADDR MsgBoxText, ADDR MsgBoxTitle, MB\_OK

invoke ExitProcess, 0

end start

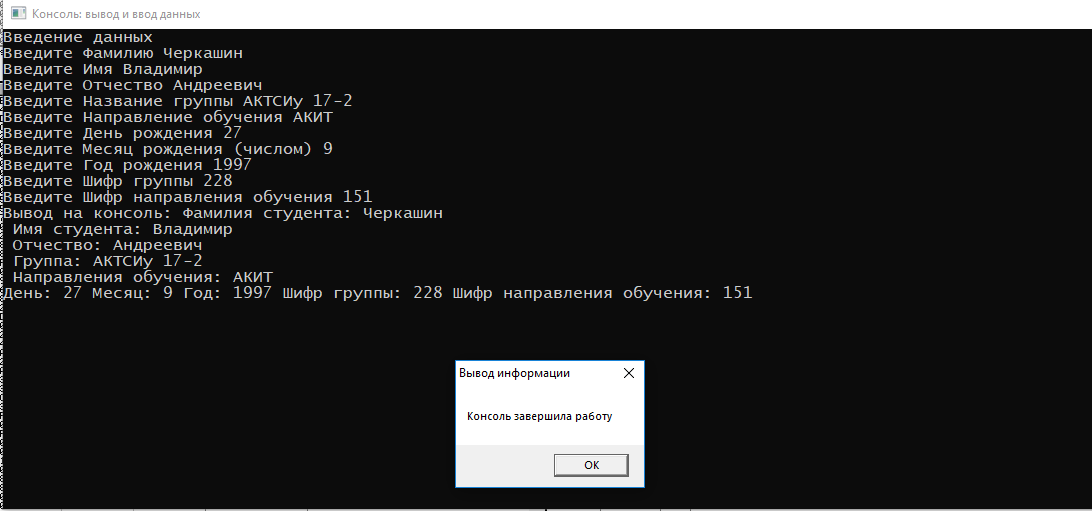


Рис 1.4 – Вывод программы задания 4

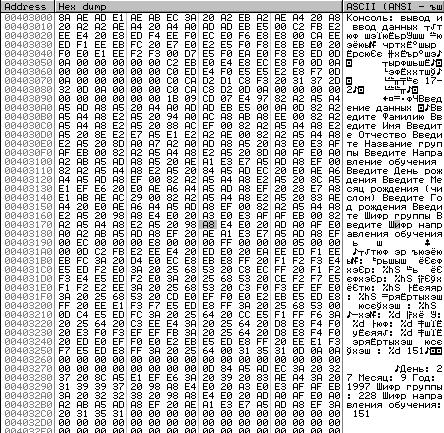


Рис 1.5 Дамп программы 3 после заполнения данными

Вывод: В ходе лабораторной работы был рассмотрен ввод и вывод данных на консоль, а так же вывод данных с помощь MessageBox, были рассмотренны особенности работы с кодировками ANSI, OEM. Так же были рассмотрены такие функции как wsprintf, CharToOEM, OemToChar и т.д. Были выучены основные директивы занесения данных в сегмент данных.