

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МО ЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №1**  
**по дисциплине «Организация ЭВМ и систем»**  
**ТЕМА: ТРАНСЛЯЦИИ, ОТЛАДКА И ВЫПОЛНЕНИЕ ПРОГРАММ НА ЯЗЫКЕ**  
**АССЕМБЛЕРА**

Студент гр. 1383

Манучарова А.С.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

## **Цель работы.**

Получение навыков трансляции, выполнения и отладки программ на языке Ассемблера.

## **Задание.**

1. Просмотреть программу `hello1.asm`, которая формирует и выводит на экран приветствие пользователя с помощью функции ОС MSDOS, вызываемой через прерывание с номером 21H (команда `Int 21h`).

Выполняемые функцией действия и задаваемые ей параметры - следующие:

- обеспечивается вывод на экран строки символов, заканчивающейся знаком "\$";
- требуется задание в регистре `ah` номера функции, равного `09h`, а в регистре `dx` - смещения адреса выводимой строки;
- используется регистр `ax` и не сохраняется его содержимое.

2. Разобраться в структуре и реализации каждого сегмента программы. Непонятные фрагменты прояснить у преподавателя. Строку-приветствие преобразовать в соответствии со своими личными данными.

3. Загрузить файл `hello1.asm` из каталога Задания в каталог `Masm`.

4. Протранслировать программу с помощью строки

```
> masm hello1.asm
```

с созданием объектного файла и файла диагностических сообщений (файла листинга).

Объяснить и исправить синтаксические ошибки, если они будут обнаружены транслятором.

Повторить трансляцию программы до получения объектного модуля.

5. Скомпоновать загрузочный модуль с помощью строки

```
> link hello1.obj
```

с созданием карты памяти и исполняемого файла `hello1.exe`.

6. Выполнить программу в автоматическом режиме путем набора строки

> hello1.exe

убедиться в корректности ее работы и зафиксировать результат выполнения в протоколе.

7. Запустить выполнение программы под управлением отладчика с помощью команды

> afd hello1.exe

Записать начальное содержимое сегментных регистров CS, DS, ES и SS. Выполнить программу в пошаговом режиме с фиксацией используемых регистров и ячеек памяти до и после выполнения каждой команды. Обычные команды выполняются по F1 (Step), а вызовы обработчиков прерываний (Int) - по F2 (StepProc), чтобы не входить внутрь обработчика прерываний. Продвижение по сегментам экранной формы отладчика выполняется с помощью клавиш F7 – F10 (up, down, left, right). Перезапуск программы в отладчике выполняется клавишей F3 (Retrieve). Выход из отладчика - по команде Quit.

Выполнить пункты 1 - 7 части 1 настоящего задания применительно к программе hello2.asm, приведенной в каталоге Задания, которая выводит на экран приветствие пользователя с помощью процедуры WriteMsg, а также использует полное определение сегментов. Сравнить результаты прогона под управлением отладчика программ hello1 и hello2 и объяснить различия в размещении сегментов.

Выполнение работы.

1. hello1.ams

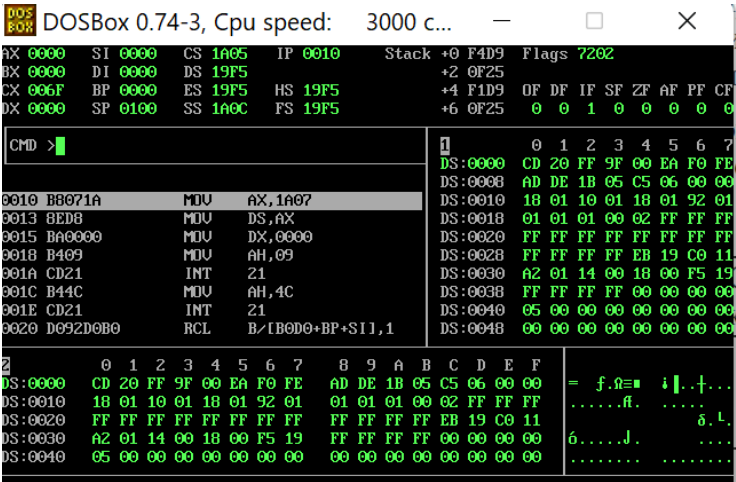


Рисунок 1 — Начальное состояние регистров hello1

Таблица 1. Протокол hello1

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0010	mov AX, 1A07	B8 07 1A	AX = 0000 IP = 0010	AX = 1A07 IP = 0013
0013	mov DS, AX	8E D8	DS = 19F5 IP = 0013	DS = 1A07 IP = 0015
0015	mov DX, 0000	BA 00 00	DX = 0000 IP = 0015	DX = 0000 IP = 0018
0018	mov AH, 09	B4 09	AX = 1A07 IP = 001A	AX = 0907 IP = 001A
001A	int 21	CD 21	IP = 001A	IP = 001C
001C	mov AH, 4C	B4 4C	AX = 0907 IP = 001C	AX = 4C07 IP = 001E
001E	int 21	CD 21	AX = 4C07 CX = 006F	AX = 0000 CX = 0000

			DS = 1A07 IP = 001E	DS = 19F5 IP = 0010
--	--	--	------------------------	------------------------

1. Просмотрен код программы hello1.asm и отредактирован в соответствии с личными данными.

2. Протранслирована программа hello1.asm с созданием объектного файла HELLO1.OBJ и файла листинга HELLO1.LST.

3. Скомпилирован загрузочный модуль HELLO1.obj с созданием файла HELLO1.EXE.

4. Выполнена программа HELLO1.EXE в автоматическом режиме: “Вас приветствует ст.гр.1383 – Манучарова А.С.”

5. Запущено выполнение программы HELLO1.EXE под управлением отладчика.

6. Начальное содержимое регистров:

CS = 1A05; DS = 19F5; ES = 19F5; SS = 1A0C

## 2. hello2.asm

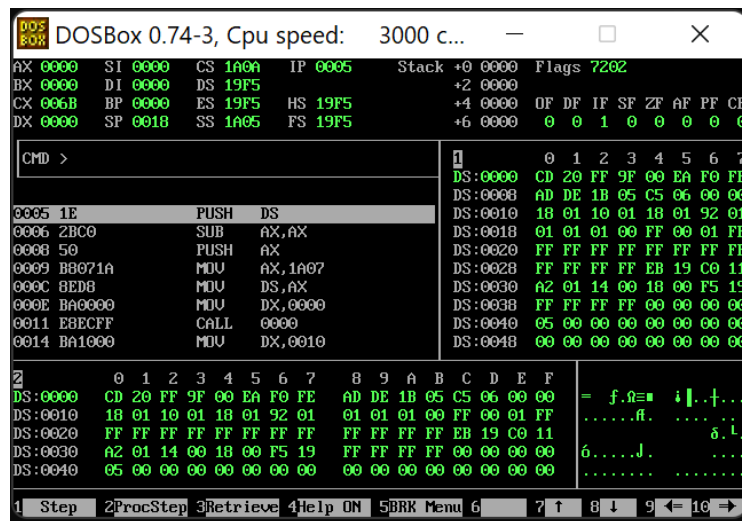


Рисунок 2 — Начальное состояние регистров hello2

Таблица 2. Протокол hello2

Адрес Команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0005	push DS	1E	IP = 0005 SP = 0018 Stack: +0 0000	IP = 0006 SP = 0016 Stack: +0 19F5
0006	sub AX,AX	2B C0	AX = 0000 IP = 0006	AX = 0000 IP = 0008
0008	push AX	50	IP = 0008 SP = 0016 Stack: +0 19F5 +2 0000	IP = 0009 SP = 0014 Stack: +0 0000 +2 19F5
0009	mov AX, 1A07	B8 07 1A	AX = 0000 IP = 0009	AX = 1A07 IP = 000C
000C	mov DS, AX	8E D8	DS = 19F5 IP = 000C	DS = 1A07 IP = 000E
000E	mov DX, 0000	BA 00 00	DX = 0000 IP = 000E	DX = 0000 IP = 0011
0011	call 0000	E8 EC FF	IP = 0011 SP = 0014 Stack: +0 0000 +2 19F5 +4 0000	IP = 0000 SP = 0012 Stack: +0 0014 +2 0000 +4 19F5
0000	mov AH, 09	B4 09	AX = 1A07 IP = 0000	AX = 0907 IP = 0002
0002	int 21	CD 21	IP = 0002	IP = 0004
0004	ret	C3	IP = 0004 SP = 0012	IP = 0014 SP = 0014

			Stack: +0 0014 +2 0000 +4 19F5	Stack: +0 0000 +2 19F5 +4 0000
0014	mov DX, 0010	BA 10 00	DX = 0000 IP = 0014	DX = 0010 IP = 0017
0017	call 0000	E8 E6 FF	IP = 0017 SP = 0014 Stack: +0 0000 +2 19F5 +4 0000	IP = 0000 SP = 0012 Stack: +0 001A +2 0000 +4 19F5
0000	mov AH, 09	B4 09	AX = 0907 IP = 0000	AX = 0907 IP = 0002
0002	int 21	CD 21	IP = 0002	IP = 0004
0004	ret	C3	IP = 0004 SP = 0012 Stack: +0 001A +2 0000 +4 19F5	IP = 001A SP = 0014 Stack: +0 0000 +2 19F5 +4 0000
001A	ret far	CB	IP = 001A SP = 0014 CS = 1A0B Stack: +0 0000 +2 19F5 +4 0000 +6 0000	IP = 0000 SP = 0018 CS = 19F5 Stack: +0 0000 +2 0000 +4 0000 +6 0000
0000	int 20	CD 20	AX = 0907	AX = 0000

1. Просмотрен код программы hello2.asm и отредактирован в соответствии с личными данными.

2. Протранслирована программа hello2.asm с созданием объектного файла hello2.exe и файла листинга hello2.lst.

3. Скомпилирован загрузочный модуль hello2.obj с созданием файла hello2.exe.

4. Выполнена программа hello2.exe в автоматическом режиме: “Hello World!\nStudent from 1383 – Manucharova A.”

5. Запущено выполнение программы hello1.exe под управлением отладчика.

6. Начальное содержимое регистров:

CS = 1A05; DS = 19F5; ES = 19F5; SS = 1A0C

### **Выводы.**

Получены навыки выполнения, трансляции и отладки программ на языке Ассемблера.

Изучены исходные коды программ hello1.asm и hello2.asm, составлены протоколы работы программ с помощью отладчика.



## ПРИЛОЖЕНИЕ А

### КОДЫ ПРОГРАММЫ

#### hello1.asm:

```
; HELLO1.ASM - упрощенная версия учебной программы
лаб.раб. N1
;
; по дисциплине "Архитектура
компьютера"
;
*****
*****
; Назначение: Программа формирует и выводит на экран
приветствие
;
; пользователя с помощью функции ДОС
"Вывод строки"
;
; (номер 09 прерывание 21h), которая:
;
; - обеспечивает вывод на экран строки
СИМВОЛОВ,
;
; заканчивающейся знаком "$";
;
; - требует задания в регистре ah номера
функции=09h,
;
; а в регистре dx - смещения адреса
выводимой
;
; строки;
;
; - использует регистр ax и не
сохраняет его
;
; содержимое.
;
*****
*****
```

```

        DOSSEG
Задание сегментов под ДОС
        .MODEL    SMALL
Модель памяти-SMALL (Малая)
        .STACK    100h
Отвести под Стек 256 байт
        .DATA
Начало сегмента данных
        Greeting LABEL BYTE
Текст приветствия
        DB 'Вас приветствует ст.гр.1383 - Манучарова
А.С.',13,10,'$'
        .CODE
сегмента кода
        mov ax, @data
DS адреса начала
        mov ds, ax
данных
        mov dx, OFFSET Greeting
dx смещения
        DisplayGreeting:
        mov ah, 9
ДОС печати строки
        int 21h
экран приветствия
        mov ah, 4ch
ДОС завершения программы

```

```

        int    21h                                ; завершение
программы и выход в ДОС
        END

```

### **hello2.asm:**

```

        ; HELLO2 - Учебная программа N2   лаб.раб.#1 по
дисциплине "Архитектура компьютера"

        ;           Программа использует процедуру для печати
строки

        ;

        ;       ТЕКСТ ПРОГРАММЫ


        EOFLine  EQU   '$'                        ; Определение символьной
константы

                                                ;       "Конец строки"


        ; Стек  программы


        ASSUME CS:CODE, SS:AStack


        AStack   SEGMENT   STACK

                        DW  12  DUP('!')           ; Отводится 12 слов
памяти

        AStack   ENDS


        ; Данные программы


        DATA     SEGMENT


        ;   Директивы описания данных

```

```

HELLO      DB 'Hello Worlds!', 0AH, 0DH,EOFLine
GREETING   DB 'Student from 1383 – Manucharova A.$'
DATA       ENDS

```

; Код программы

```

CODE       SEGMENT

```

; Процедура печати строки

```

WriteMsg   PROC   NEAR

```

```

            mov    AH,9

```

```

            int     21h      ; Вызов функции DOS по

```

прерыванию

```

            ret

```

```

WriteMsg   ENDP

```

; Головная процедура

```

Main       PROC   FAR

```

```

            push   DS      ;\ Сохранение адреса начала

```

PSP в стеке

```

            sub     AX,AX      ; > для последующего

```

восстановления по

```

            push   AX      ;/ команде ret, завершающей

```

процедуру.

```

            mov     AX,DATA      ; Загрузка

```

сегментного

```

            mov     DS,AX      ; регистра данных.

```

```

            mov     DX, OFFSET HELLO ; Вывод на экран

```

первой

```

        call    WriteMsg                ; строки
приветствия.

        mov     DX, OFFSET GREETING ; Вывод на экран
второй

        call    WriteMsg                ; строки
приветствия.

        ret                                ; Выход в DOS по
команде,

                                ; находящейся в
1-ом слове PSP.

Main      ENDP
CODE      ENDS

        END Main

```

## ПРИЛОЖЕНИЕ В

### ЛИСТИНГИ ПРОГРАММ

#### hello1.lst:

```
__Microsoft (R) Macro Assembler Version 5.10  
9/27/22 21:41:40
```

Symbols-1

Segments and Groups:

	N a m e	Length	Align
Combine Class			
DGROUP . . . . .		GROUP	
__DATA . . . . .		004FWORD PUBLIC	
'DATA'			
STACK . . . . .		0100 PARA STACK	
'STACK'			
__TEXT . . . . .		0010WORD PUBLIC	
'CODE'			

Symbols:

	N a m e	Type	Value	Attr
DISPLAYGREETING . . . . .		L NEAR		0008
__TEXT				

```

GREETING . . . . . L BYTE 0000
_DATA

@CODE . . . . . TEXT _TEXT

@CODESIZE . . . . . TEXT 0
@CPU . . . . . TEXT 0101h

@DATASIZE . . . . . TEXT 0
@FILENAME . . . . . TEXT hello1

@VERSION . . . . . TEXT 510

```

```

33 Source Lines
33 Total Lines
19 Symbols

```

47994 + 459266 Bytes symbol space free

```

0 Warning Errors
0 Severe Errors

```

# **hello2.lst:**

```

__Microsoft (R) Macro Assembler Version 5.10
9/27/22 21:44:28

```

Symbols-1

# Segments and Groups:

N a m e	Length	Align
Combine Class		
ASTACK . . . . .	0018	PARA STACK
CODE . . . . .	001B	PARA NONE
DATA . . . . .	0025	PARA NONE

## Symbols:

N a m e	Type	Value	Attr
EOFLINE . . . . .	NUMBER		0024
GREETING . . . . .	L BYTE		0010
DATA			
HELLO . . . . .	L BYTE		0000
DATA			
MAIN . . . . .	F PROC		0005
CODELength = 0016			
WRITEMSG . . . . .	N PROC		0000
CODELength = 0005			



@CPU . . . . .	TEXT	0101h
@FILENAME . . . . .	TEXT	hello2
@VERSION . . . . .	TEXT	510

52 Source Lines

52 Total Lines

13 Symbols

47994 + 459263 Bytes symbol space free

0 Warning Errors

0 Severe Errors