

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ  
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ «ЛЭТИ»  
ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)»  
Кафедра МО ЭВМ**

**ОТЧЁТ  
по лабораторной работе № 1  
по дисциплине «Организация ЭВМ и систем»  
Тема: Трансляция, отладка и выполнение программ на языке  
Ассемблера**

Студент гр. 1303

Преподаватель

Карагезов С.Ю

Ефремов М.А.

Санкт-Петербург

2022

### **Цель работы.**

Изучить структуру и работу простейших программ, написанных на языке Ассемблера. Получить навыки в работе с транслятором и отладчиком.

### **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. 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 компилятором.

Повторить трансляцию программы для получения объектного модуля.

5. Скомпоновать загрузочный модуль с помощью строки

> link hello1.obj

с созданием карты памяти исполняемого файла hello1.exe.

6. Выполнить программу в автоматическом режиме путём набора строки

> hello1.exe

убедиться в корректности её 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.

### **Выполнение работы.**

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

2. Протранслирована программа hello1.asm, в результате которого был получен объектный файл hello1.obj и файл листинга hello1.lst.

3. Скомпонован загрузочный модуль hello1.obj, в результате которого были получены карта памяти hello1.map и исполняемый файл hello1.exe.

4. Выполнена программа hello1.exe в автоматическом режиме: "Вас приветствует ст.гр.1303 - Карагезов С.Ю."

5. Под управлением отладчика была запущена программа hello1.exe.

6. Начальное содержимое регистров:

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

Таблица 1 – Результат прогона программы hello2.exe в отладчике.

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) = 0018	(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) = 006B (DS) = 1A07 (IP) = 001E	(AX) = 0000 (CX) = 0000 (DS) = 19F5 (IP) = 0010

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

2. Протранслирована программа hello2.asm, в результате которого был получен объектный файл hello2.obj и файл листинга hello2.lst.

3. Скомпонован загрузочный модуль hello2.obj, в результате которого были получены карта памяти hello2.map и исполняемый файл hello2.exe.

4. Была запущена программа hello2.exe в автоматическом режиме:  
"Hello World!\nStudent from 1303 - Karagezov S.Y"

5. Под управлением отладчика была запущена программа hello2.exe.

6. Начальное содержимое регистров:

CS = 1A0B; DS = 19F5; ES = 19F5; SS = 1A05;

Таблица 2 – Результат прогона программы hello2.exe в отладчике.

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	(IP) = 0006 (AX) = 0000	(IP) = 0008 (AX) = 0000
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	(IP) = 0009 (AX) = 0000	(IP) = 000C (AX) = 1A07
000C	mov DS,AX	8E D8	(IP) 000C (DS) = 19F5	(IP) = 000E (DS) = 1A07
000E	mov DX,0000	BA 00 00	(IP) = 000E (DX) = 0000	(IP) = 0011 (DX) = 0000
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	(IP) = 0000 (AX) = 1A07	(IP) = 0002 (AX) = 0907
0002	int 21	CD 21	(IP) = 0002	(IP) = 0004

0004	ret	C3	(IP) = 0004 (SP) = 0012 Stack: +0 0014 +2 0000 +4 19F5	(IP) = 0014 (SP) = 0014 Stack: +0 0000 +2 19F5 +4 0000
0014	mov DX,0010	BA 10 00	(IP) = 0014 (DX) = 0000	(IP) = 0017 (DX) = 0010
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	(IP) = 0000 (AX) = 0907	(IP) = 0002 (AX) = 0907
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 (CS) = 1A0A (SP) = 0014 Stack: +0 0000 +2 19F5	(IP) = 0000 (CS) = 19F5 (SP) = 0018 Stack: +0 0000 +2 0000

0000	int 20	CD 20	(IP) = 0000 (CS) = 19F5 (AX) = 0907 (CX) = 007B (DX) = 0010 (DS) = 1A07	(IP) = 0005 (CS) = 1A0A (AX) = 0000 (CX) = 0000 (DX) = 0000 (DS) = 19F5
------	--------	-------	--	--

### **Выводы**

Изучена структура и работа простейших программ, написанных на языке Ассемблера. Получены навыки в работе с транслятором и отладчиком.

# ПРИЛОЖЕНИЕ А

## ЛИСТИНГИ ПРОГРАММ

Название файла: hello1.lst

Microsoft (R) Macro Assembler Version 5.10

12/4/22 01:57:07

Page 1-1

```

DOSSEG
.MODEL SMALL
.STACK 100h
.DATA
0000      Greeting LABEL BYTE
0000  D0 92 D0 B0 D1 81      DB 'Вас приветствует ст.г
                                ?'.1303 - Карагезов С.Ю.',13,10,'$'
                                ?
20 D0 BF D1 80 D0
B8 D0 B2 D0 B5 D1
82 D1 81 D1 82 D0
B2 D1 83 D0 B5 D1
82 20 D1 81 D1 82
2E D0 B3 D1 80 2E
31 33 30 33 20 2D
20 D0 9A D0 B0 D1
80 D0 B0 D0 B3 D0
B5 D0 B7 D0 BE D0
B2 20 D0 A1 2E D0
AE 2E 0D 0A 24

.CODE
0000  B8 ---- R      mov ax, @data
0003  8E D8          mov ds, ax
0005  BA 0000 R      mov dx, OFFSET Greeting

0008      DisplayGreeting:
0008  B4 09          mov ah, 9
000A  CD 21          int 21h
000C  B4 4C          mov ah, 4ch
000E  CD 21          int 21h
END

```

Microsoft (R) Macro Assembler Version 5.10

12/4/22 01:57:07

Symbols-1

Segments and Groups:

N a m e	Length	Align	Combine	Class
DGROUP . . . . .	GROUP			
_DATA . . . . .	004D	WORD	PUBLIC	'DATA'
STACK . . . . .	0100	PARA	STACK	'STACK'



```
_TEXT . . . . . 0010 WORD 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	

```
17 Source Lines
17 Total Lines
19 Symbols
```

47994 + 463361 Bytes symbol space free

```
0 Warning Errors
0 Severe Errors
```

Название файла: hello2.lst

```
Microsoft (R) Macro Assembler Version 5.10      12/4/22 01:57:12
                                           Page    1-1
```

```
= 0024                      EOFLine EQU '$'
```

```
ASSUME CS:CODE, SS:AStack
```

```
0000                      AStack  SEGMENT STACK
0000 000C[                  DW 12 DUP('!')
    0021                      ]
```

```
0018                      AStack  ENDS
```

```
0000                      DATA   SEGMENT
0000 48 65 6C 6C 6F 20      HELLO   DB 'Hello World!', 0AH, 0DH, EOFLine
    57 6F 72 6C 64 21
    0A 0D 24
000F 53 74 75 64 65 6E      GREETING DB 'Student from 1303 - Karagezov
    S.Y.$'
```

```

    74 20 66 72 6F 6D
    20 31 33 30 33 20
    2D 20 4B 61 72 61
    67 65 7A 6F 76 20
    53 2E 59 2E 24
0032                                DATA    ENDS

0000                                CODE     SEGMENT

0000                                WriteMsg PROC NEAR
0000 B4 09                          mov     AH,9
0002 CD 21                          int     21h
0004 C3                             ret
0005                                WriteMsg ENDP

0005                                Main     PROC FAR
0005 1E                             push    DS
0006 2B C0                          sub     AX,AX
0008 50                             push    AX
0009 B8 ---- R                      mov     AX,DATA
000C 8E D8                          mov     DS,AX
000E BA 0000 R                      mov     DX, OFFSET HELLO
0011 E8 0000 R                      call    WriteMsg
0014 BA 000F R                      mov     DX, OFFSET GREETING
0017 E8 0000 R                      call    WriteMsg
001A CB                             ret
001B                                Main     ENDP

001B                                CODE     ENDS
                                END Main
Microsoft (R) Macro Assembler Version 5.10                12/4/22 01:57:12
                                Symbols-1

```

#### Segments and Groups:

N a m e	Length	Align	Combine Class
ASTACK . . . . .	0018	PARA	STACK
CODE . . . . .	001B	PARA	NONE
DATA . . . . .	0032	PARA	NONE

#### Symbols:

N a m e	Type	Value	Attr
EOFLINE . . . . .	NUMBER	0024	
GREETING . . . . .	L BYTE	000F	DATA
HELLO . . . . .	L BYTE	0000	DATA

MAIN . . . . .	F PROC	0005	CODE	Length = 0016
WRITEMSG . . . . .	N PROC	0000	CODE	Length = 0005
@CPU . . . . .	TEXT	0101h		
@FILENAME . . . . .	TEXT	hello2		
@VERSION . . . . .	TEXT	510		

36 Source Lines

36 Total Lines

13 Symbols

48002 + 463353 Bytes symbol space free

0 Warning Errors

0 Severe Errors

## ПРИЛОЖЕНИЕ В

### КОД ПРОГРАММ

Название файла: hello1.asm

```
DOSSEG
.MODEL SMALL
.STACK 100h
.DATA
Greeting LABEL BYTE
DB 'Вас приветствует ст.гр.1303 - Карагезов С.Ю.',13,10,'$'
.CODE
mov ax, @data
mov ds, ax
mov dx, OFFSET Greeting

DisplayGreeting:
mov ah, 9
int 21h
mov ah, 4ch
int 21h
END
```

Название файла: hello2.asm

```
EOFLine EQU '$'

ASSUME CS:CODE, SS:AStack

AStack SEGMENT STACK
    DW 12 DUP('!')
AStack ENDS

DATA SEGMENT
    HELLO DB 'Hello World!', 0AH, 0DH, EOFLine
    GREETING DB 'Student from 1303 - Karagezov S.Y.$'
DATA ENDS

CODE SEGMENT

WriteMsg PROC NEAR
    mov AH,9
    int 21h
    ret
WriteMsg ENDP

Main PROC FAR
    push DS
    sub AX,AX
```

```
    push AX
    mov  AX,DATA
    mov  DS,AX
    mov  DX, OFFSET HELLO
    call WriteMsg
    mov  DX, OFFSET GREETING
    call WriteMsg
    ret
Main    ENDP

CODE    ENDS
        END Main
```