

**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. 9382

\_\_\_\_\_

Бочаров Г.С.

Преподаватель

\_\_\_\_\_

Ефремов М.А.

Санкт-Петербург

2020

## **Цель работы.**

Изучить основные принципы трансляции, отладки и выполнения программ на языке Ассемблера.

## **Ход работы.**

### **Часть 1.**

1. Загрузка файлов AFDPRO.EXE, HELLO1.ASM, HELLO2.ASM, LINK.EXE, MASM.EXE в один каталог.
2. Запуск DOSBox, создание виртуального диска E.
3. Просмотр кода программы. Были изучены принципы работы каждого сегмента. Преобразовал строку-приветствие.
4. Выполнил трансляцию программы hello1.asm с помощью masm.exe. В результате трансляции был создан объектный файл hello1.obj и файл листинга hello1.lst. Трансляция прошла без ошибок и предупреждений.
6. Слинковал загрузочный модуль hello1.exe с помощью link.exe. В результате работы линковщика создался загрузочный модуль hello1.exe.
8. Запуск программы в автоматическом режиме путем набора строки:  
    > hello1.exe
9. Вывод программы:  
    > Вас приветствует ст.гр. 9382 – Бочаров Г.С..
10. Используя отладчик, выполнил запуск программы hello1.exe. В ходе выполнения программы записывались используемые регистры и ячейки памяти до и после выполнения команд. Отладчик был запущен при помощи команды:  
    > afdpro.exe

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.

```
AX 0000 SI 0000 CS 1D77 IP 0010
BX 0000 DI 0000 DS 1D67
CX 004D BP 0000 ES 1D67 HS 1D67
DX 0000 SP 0100 SS 1D7C FS 1D67
```

Адрес Команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			до выполнения	После выполнения
0010	MOV AX, 1D79	B8791D	(AX) = 0000 (IP) = 0010	(AX) = 1D79 (IP) = 0013
0013	MOV DS, AX	8ED8	(AX) = 1D79 (DS) = 1D67 (IP) = 0013	(AX) = 1D79 (DS) = 1D79 (IP) = 0015
0015	MOV DX, 0000	BA0000	(DX) = 0000 (IP) = 0015	(DX) = 0000 (IP) = 0018
0018	MOV AH,09	B409	(AX) = 1D79 (IP) = 0018	(AX) = 0979 (IP) = 001A
001A	INT 21	CD21	(IP) = 001A	(IP) = 001C
001C	MOV AH,4C	B44C	(AX) = 0979 (IP) = 001C	(AX) = 4C79 (IP) = 001E
001E	INT 21	CD21	(AX) = 4C79 (DS) = 1D79 (IP) = 001E	(AX) = 0000 (DS) = 1D67 (IP) = 0010

## Часть 2.

1. Просмотрел программу hello1.ASM в текстовом редакторе, изучил структуру и реализации каждого сегмента программы. Преобразовал строку-приветствие.

2. Выполнил трансляцию программы hello2.ASM с помощью транслятора MASM и команды:

```
>masm hello2.asm
```

В результате трансляции был создан объектный файл hello2.obj и файл листинга hello1.lst. Трансляция прошла без ошибок, но с одним предупреждением.

3. Используя линковщик LINK, создал загрузочный модуль hello2.EXE.

Использованная команда:

>link hello2.exe

4. Запустил программу hello2.exe и проверил корректность ее работы:

```
E:\LB_1>hello2.exe
Приветствую
Студент Бочаров Г.С. группы 9382
E:\LB_1>
```

5. Запустил программу с помощью отладчика AFDPRO.EXE и пошагово исполнил с записью используемых регистров и ячеек команд.

```
AX 0000 SI 0000 CS 1D7C IP 0005
BX 0000 DI 0000 DS 1D67
CX 006B BP 0000 ES 1D67 HS 1D67
DX 0000 SP 0018 SS 1D77 FS 1D67
```

Табл.2.

Адрес Команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0005	PUSH DS	1E	(SP) = 0018 (IP) = 0005 Stack +0 0000 +2 0000 +4 0000 +6 0000	(SP)=0016 (IP) = 0006 Stack +0 1D67 +2 0000 +4 0000 +6 0000
0006	SUB AX, AX	2BC0	(AX) = 0000 (IP) = 0006	(AX) = 0000 (IP) = 0008
0008	PUSH AX	50	(AX) = 0000 (SP)=0016 (IP) = 0008 Stack +0 1D67 +2 0000 +4 0000	(AX) = 0000 (SP) = 0014 (IP) = 0009 Stack +0 0000 +2 1D67 +4 0000
0009	MOV AX, 1D79	B8791D	(AX) = 0000 (IP) = 0009	(AX) = 1D79 (IP) = 000C
000C	MOV DS, AX	8ED8	(AX) = 1D79 (DS) = 1D67 (IP) = 000C	(AX) = 1D79 (DS) = 1D79 (IP) = 000E

000E	MOV DX, 0000	BA0000	(DX) = 0000 (IP) = 000E	(DX) = 0000 (IP) = 0011
0011	CALL 0000	E8ECFF	(SP) = 0014 (IP) = 0011 Stack +0 0000 +2 1D67 +4 0000	(SP) = 0012 (IP) = 0000 Stack +0 0014 +2 0000 +4 1D67
0000	MOV AH,09	B409	(AX) = 1D79 (IP) = 0000	(AX) = 0979 (IP) = 0002
0002	INT 21	CD21	(IP) = 0002	(IP) = 0004
0004	RET	C3	(SP) = 0012 (IP) = 0004 Stack +0 0014 +2 0000 +4 1D67	(SP) = 0014 (IP) = 0014 Stack +0 0000 +2 1D67 +4 0000
0014	MOV DX,000E	BA0E00	(DX) = 0000 (IP) = 0014	(DX) = 000E (IP) = 0017
0017	CALL 0000	E8E6FF	(SP) = 0014 (IP) = 0017 Stack +0 0000 +2 1D67 +4 0000	(SP) = 0012 (IP) = 0000 Stack +0 001A +2 0000 +4 1D67
0000	MOV AH,09	B409	(AX) = 0979 (IP) = 0000	(AX) = 0979 (IP) = 0002
0002	INT 21	CD21	(IP) = 0002	(IP) = 0004
0004	RET	C3	(SP) = 0012 (IP) = 0004 Stack +0 001A +2 0000 +4 1D67	(SP) = 0014 (IP) = 001A Stack +0 0000 +2 1D67 +4 0000
001A	RET Far	CB	(SP) = 0014 (IP) = 001A (CS) = 1D7C Stack +0 0000 +2 1D67 +4 0000	(SP) = 0018 (IP) = 0000 (CS) = 1D67 Stack +0 0000 +2 0000 +4 0000
0000	INT 20	CD20	(AX) = 0979 (DS) = 1D79 (IP) = 0000	(AX) = 0000 (DS) = 1D67 (IP) = 0005

### **Выводы.**

В результате выполнения лабораторной работы был освоен этап трансляции, а также выполнение и пошаговое исполнение программы при помощи отладчика на языке Ассемблера.



## Приложение А

### Текст файла HELLO1.ASM

```
                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.9382 Бочаров Глеб',13,10,'$'
                .CODE                                ; Начало сегмента кода
                mov  ax, @data                        ; Загрузка в DS адреса
начала
                mov  ds, ax                          ; сегмента данных
                mov  dx, OFFSET Greeting              ; Загрузка в dx смещения
                                                    ; адреса текста
приветствия
                DisplayGreeting:
                mov  ah, 9                            ; # функции ДОС печати
строки
                int  21h                              ; вывод на экран
приветствия
                mov  ah, 4ch                          ; # функции ДОС
завершения программы
                int  21h                              ; завершение программы и
выход в ДОС
                END
```

### Текст файла HELLO1.LST

```
#Microsoft (R) Macro Assembler Version 5.10
1/9/21 06:07:12
```

Page 1-1

~à£à ; HELLO1.ASM - ã~à®é¥ ĭ ç¥àášĭ ãç¥ĭ®©

```

      ¬¬ë « j.à j. N1
;      ® €šáæš¬«š¥
"□àâšâ¥ªâãà  ª®¬¬
      ìîâ¥à "
;
*****
*****
;  □ š ç¥š¥: □à®£à ¬¬  ä®à¬šâã¥â š çë
ç®€šâ
      íªà  ¬àšç¥âââçš¥
;      ®«ì§®ç  â¥«î á  ®¬®éîî äã-
ªæšš  □□
      □ "□ëç®€  áâà®ªš"
;      (®¬¥à 09  ¬à¥àëç  š¥ 21h),
ª®â®à
      ì:
;      - ®j¥á¬¥çšç  ¥â çëç®€
      íªà  áâ
      à®ªš  áš¬ç®«®ç,
;      §  ª  çšç  îé¥©áî §  ª®¬
"$";
;      - âà¥jã¥â §  €  šî ç  à¥
£šáâà¥ ah
      ®¬¥à  äãªæšš=09h,
;      ç  à¥£šáâà¥ dx -
      á¬¥é¥šî
      €à¥á  çëç®€š¬®©
;      áâà®ªš;
;      - šá¬®«ì§ã¥â  à¥£šáâà  ax
š  ¥
      á®âà  î¥â  ¥£®
;      á®€¥àšš¬®¥.
;
*****
*****

```

```

      DOSSEG
;  □ €  š¥  á¥£¬¥â®ç  ®®  □□□
      .MODEL  SMALL
;  □®€¥«ì  ¬¬îâš-SMALL(□ « î)
      .STACK  100h
;  □âç¥áâš  ®®  □â¥ª  256  j  ©â
      .DATA
;  □ ç  «®  á¥£¬¥â  €  ëâ
      Greeting LABEL BYTE

```

0000



```

;  ¤¥ªªª ¤àš¤¥ªªª¤šĩ
0000 82 A0 E1 20 AF E0 DB ' ¤ ¤ ¤àš¤¥ªªª¤šªª ¤ª.£à.9382
- ¤ª¤ ¤ª¤ ¤.
    ¤. ',13,10, '$'
A8 A2 A5 E2 E1 E2
A2 E3 A5 E2 20 E1
E2 2E A3 E0 2E 39
33 38 32 20 2D 20
81 AE E7 A0 E0 AE
A2 20 83 2E 91 2E
0D 0A 24
.CODE
; ¤ ¤ ¤ ¤
    ¤ ¤ª£¬¬ªª ¤ª¤
0000 B8 ---- R      mov ax, @data
; ¤ £à
    ¤ªª ¤ ¤ DS ¤àªª ¤ ¤ ¤
0003 8E D8      mov ds, ax
; ¤ª£¬¬
    ¤ª ¤ ¤ ¤ª
0005 BA 0000 R      mov dx, OFFSET Greeting
; ¤ £à
    ¤ªª ¤ ¤ dx ¤¬ªéªšĩ

```

Page 1-2

```
;  €à¥á
                                â¥ªâ ¯àš¥âââçšï
                                DisplayGreeting:
                                0008      mov  ah, 9
                                0008  B4 09
; #  äã
                                ªæšš □□ ¯¥ç âš áâàªªš
                                000A  CD 21      int  21h
; çëçª€
                                íªà ¯àš¥âââçšï
                                000C  B4 4C      mov  ah, 4ch
; #  äã
                                ªæšš □□ § ç¥àè¥šï ¯àª£à ¬¬ë
                                000E  CD 21      int  21h
; § ç¥à
                                è¥š¥ ¯àª£à ¬¬ë š çëªª€ ç □□□
                                END
```

```
#Microsoft (R) Macro Assembler Version 5.10
2/9/21 06:07:12
```

## Symbols-1

### Segments and Groups:

Class	Name	Length	Align	Com-
DGROUP	. . . . .	GROUP		
_DATA	. . . . .	002D WORD PUBLIC		
'DATA'				
STACK	. . . . .	0100 PARA STACK		
'STACK'				
_TEXT	. . . . .	0010 WORD PUBLIC		
'CODE'				

### Symbols:

Name	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
```

47970 + 447257 Bytes symbol space free

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

### ***Текст файла HELLO2.ASM***

EOFLine EQU '\$'

AStack SEGMENT STACK

DW 12 DUP(?)

AStack ENDS

DATA SEGMENT

HELLO DB 'Приветствую', 0AH, 0DH, EOFLine

GREETING DB 'Студент Бочаров Г.С. группы 9382 - \$'

DATA ENDS

CODE SEGMENT

ASSUME CS:Code DS:DATA SS:AStack

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

```

## Текст файла HELLO2.LST

#Microsoft (R) Macro Assembler Version 5.10  
07:00:23

1/9/21

Page

1-1

```

; HELLO2 - 0x7i i à@£à ¬¬ N2 « i.à i.#1
;
€šáæš¬«š¥ "0àâšâ¥ªââà ª@¬îîâ¥à "
;
€«î ¬
;
¥ç âš áâàªš
;
;
= 0024 EOFLine EQU '$' ; 0¬à¥€¥«¥š¥
áš¬ç@«î
©ª ª@ââ âë
;
"0¥æ
áâàªš"
;
; 0â¥ª ¬à@£à ¬¬ë
0000 AStack SEGMENT STACK
0000 000C[ DW 12 DUP(?) ; 0âç@€šââî
12 á«@ç ¬
¬îâš
????
]
0018 AStack ENDS

```



1-2

```

0006  2B C0          sub     AX,AX      ; > €«i ¯@á«¥
€ãíé¥£® ¢

0008  50            push    AX          ;/ ¢@¬ €¥ ret, § ¢
¥à

0009  B8 ---- R     mov     AX,DATA      ;
¤ £ãã§ª

000C  8E D8          mov     DS,AX          ;
à¥£šáâà

000E  BA 0000 R     mov     DX, OFFSET HELLO ; ¢ë¢®€

0011  E8 0000 R     call    WriteMsg      ; áâàªªš
¯à

0014  BA 000E R     mov     DX, OFFSET GREETING ; ¢ë¢®€

0017  E8 0000 R     call    WriteMsg      ; áâàªªš
¯à

001A  CB            ret                    ; ¢ëâ®€
¢ D

                                OS ¯® ¢@¬ €¥,
                                ; -

â®€íé¥©

001B                                áĩ ¢ 1-®¬ á«®¢¥ PSP.
Main                                ENDP
001B                                CODE                                ENDS
                                END Main

```

```
#Microsoft (R) Macro Assembler Version 5.10
2/9/21 07:00:23
```

Symbols-1

Segments and Groups:

Class	N a m e	Length	Align	Combine
ASTACK	. . . . .	0018	PARA	STACK
CODE	. . . . .	001B	PARA	NONE
DATA	. . . . .	0030	PARA	NONE

Symbols:

	N a m e	Type	Value	Attr
EOFLINE	. . . . .	NUMBER		0024
GREETING	. . . . .	L BYTE		000E 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	

```
51 Source Lines
51 Total Lines
13 Symbols
```

47978 + 447249 Bytes symbol space free

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