

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 Ассем-
блера.

Студентка гр. 9383

Лихашва А.Д.

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. Загружены файлы HELLO1.asm, HELLO2.asm, masm.exe, link.exe, afdpro.exe из каталога \лр в каталог D:\asmlab.
2. Запущена программа DOSBox, смонтирован виртуальный диск C: в каталоге \asmlab при помощи mount C D:\asmlab .
3. Просмотрена программа в режиме редактирования, изучена структура и реализация каждого сегмента программы. Строка-приветствие преобразована в соответствии с личными данными.
4. В DOS осуществлен переход на виртуальный диск при помощи команды C: cd D:\asmlab
5. Протранслирована программа с помощью строки:
 > MASM HELLO1.asm
По ходу трансляции создается объектный файл HELLO1.obj. Во время её выполнения ошибок не было обнаружено.
6. Скомпонован загрузочный модуль Hello1.exe с помощью строки:
 > LINK HELLO1.obj
В результате работы линковщика создается загрузочный модуль HELLO1.exe.
7. Загружена русская кодовая таблица символов путём набора строки:
 > keyb ru 866
8. Запущена программа в автоматическом режиме путем набора строки:
 > HELLO1.exe
9. Вывод программы:

> Hello! Im Lihashva Anastasia from 9383

10. Выполнен запуск программы HELLO1.exe в пошаговом режиме с фиксацией используемых регистров и ячеек памяти до и после выполнения каждой команды, используя отладчик и соответственно команду:

> AFDPRO HELLO1.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.

Содержимое сегментных регистров до старта программы: CS:1A05, DS:19F5, ES:19F5, SS:1A0A, HS:19F5, FS:19F5

Адрес Команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			до выполнения .	После выполнения
0010	MOV AX, 1A07	B8071A	(AX) = 0000 (DS) = 19F5 (DX) = 0000 (IP) = 0010	(AX) = 1A07 (DS) = 19F5 (DX) = 0000 (IP) = 0013
0013	MOV DS, AX	8ED8	(AX) = 1A07 (DS) = 19F5 (DX) = 0000 (IP) = 0013	(AX) = 1A07 (DS) = 1A07 (DX) = 0000 (IP) = 0015
0015	MOV DX, 0000	BA0000	(AX) = 1A07 (DS) = 1A07 (DX) = 0000 (IP) = 0015	(AX) = 1A07 (DS) = 1A07 (DX) = 0000 (IP) = 0018
0018	MOV AH,09	B409	(AX) = 1A07 (DS) = 1A07 (DX) = 0000 (IP) = 0018	(AX) = 0907 (DS) = 1A07 (DX) = 0000 (IP) = 001A
001A	INT 21	CD21	(AX) = 0907 (DS) = 1A07 (DX) = 0000 (IP) = 001A	(AX) = 0907 (DS) = 1A07 (DX) = 0000 (IP) = 001C
001C	MOV AH,4C	B44C	(AX) = 0907 (DS) = 1A07 (DX) = 0000 (IP) = 001C	(AX) = 4C07 (DS) = 1A07 (DX) = 0000 (IP) = 001E
001E	INT 21	CD21	(AX) = 4C07 (DS) = 1A07 (DX) = 0000 (IP) = 001E	(AX) = 0000 (DS) = 19F5 (DX) = 0000 (IP) = 0010

Листинг:

```

DOSSEG

; ҝ ҝ ЁГ ЪГҜ-ГВ®Ў Ё®  „Ҝ\
.MODEL SMALL

; Ҝ®ҝГ«м Ё -пвЁ-SMALL(Ҝ « п)

.STACK 100h

; ҜвЎГЪвЁ Ё® \вГЄ 256 Ў ©в

.DATA

; Ҝ з «® ЪГҜ-ГВ ҝ ле
0000 Greeting LABEL BYTE

; 'ГЄвв ЁаЁЎГвбвЎЁп
0000 48 65 6C 6C 6F 21 DB 'Hello! Im Lihashva Anastasia from
9383',

13,10,'$'

20 49 6D 20 4C 69
68 61 73 68 76 61
20 41 6E 61 73 74
61 73 69 61 20 66
72 6F 6D 20 39 33
38 33 0D 0A 24

.CODE ;

Ҝ з «

® ЪГҜ-ГВ €®

0000 B8 ---- R mov ax, @data

; ҝ Ja

ГЄЄ Ў DS ҝаГЪ з «

0003 8E D8 mov ds, ax

; ЪГҜ-

ГВ ҝ ле

0005 BA 0000 R mov dx, OFFSET Greeting

; ҝ Ja

ГЄЄ Ў dx Ъ-ГйГЁп

;

ҝаГЪ

```

```

                                вГЄбв İaĖŸГбвŸĖп
0008                                DisplayGreeting:
0008  B4 09                                mov  ah, 9
; # дГ

                                ЄжĖĖ „Г` İГз вĖ бва®ЄĖ
000A  CD 21                                int
21h                                ; ŸлŸ®
                                нЄa İaĖŸГбвŸĖп
000C  B4 4C                                mov  ah, 4ch
; # дГ

                                ЄжĖĖ „Г` $ ŸГаиГĖп İa®Ja ¬¬л
000E  CD 21                                int
21h                                ; $ ŸГа
                                иГĖГ İa®Ja ¬¬л Ė Ÿле® Ÿ „Г`
                                END

```

Symbols-1

Segments and Groups:

	N a m e	Length	Align	Combine
Class				
DGROUP	GROUP		
_DATA	0029 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
```

```
33 Source  Lines
```

```
33 Total   Lines
```

```
19 Symbols
```

```
47998 + 461309 Bytes symbol space free
```

Часть 2.

1. Просмотрена программа HELLO2.asm в режиме редактирования, изучена ее структура и реализация каждого сегмента программы. Строки-приветствия преобразованы в соответствии с личными данными.

2. Выполнена трансляция программы HELLO2.asm с помощью транслятора MASM и команды:

```
>MASM HELLO2.asm
```

В результате чего получился объектный файл HELLO2.obj. В процессе трансляции ошибок не обнаружено.

3. Используя линковщик LINK, создан загрузочный модуль HELLO2.exe. Использована команда:

```
>LINK HELLO2.obj
```

4. Выполнена программа HELLO2.exe в автоматическом режиме и проконтролировано, что она работает корректно: в консоль выводится:

```
>Hello World!
```

```
>Lihashva Anastasia from 9383.
```

5. Запущена программа HELLO2.exe в пошаговом режиме, используя отладчик AFDPRO с фиксацией содержимого используемых регистров и ячеек памяти до и после выполнения команд.

Содержимое сегментных регистров до старта программы: CS:1A08, DS:19F5, ES:19F5, SS:1A05, HS:19F5, FS: 19F5.

Адрес Команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0005	PUSH DS	1E	(AX) = 0000 (DS) = 19F5 (SP) = 0000 (IP) = 0005 Stack +0 0000 +2 0000 +4 0000	(AX) = 0000 (DS) = 19F5 (SP) = FFFE (IP) = 0006 Stack +0 19F5 +2 6548 +4 6C6C
0006	SUB AX, AX	2BC0	(AX) = 0000 (DS) = 19F5 (SP) = FFFE (IP) = 0006 Stack +0 19F5 +2 6548 +4 6C6C	(AX) = 0000 (DS) = 19F5 (SP) = FFFE (IP) = 0008 Stack +0 19F5 +2 6548 +4 6C6C
0008	PUSH AX	50	(AX) = 0000 (DS) = 19F5 (SP) = FFFE (IP) = 0008 Stack +0 19F5 +2 6548 +4 6C6C	(AX) = 0000 (DS) = 19F5 (SP) = FFFC (IP) = 0009 Stack +0 0000 +2 19F5 +4 6548
0009	MOV AX, 1A05	B8071A	(AX) = 0000 (DS) = 19F5 (SP) = FFFC (IP) = 0009 Stack +0 0000 +2 19F5 +4 6548	(AX) = 1A05 (DS) = 19F5 (SP) = FFFC (IP) = 000C Stack +0 0000 +2 19F5 +4 6548
000C	MOV DS, AX	8ED8	(AX) = 1A05 (DS) = 19F5 (SP) = FFFC (IP) = 000C Stack +0 0000 +2 19F5 +4 6548	(AX) = 1A05 (DS) = 1A05 (SP) = FFFC (IP) = 000E Stack +0 0000 +2 19F5 +4 6548

000E	MOV DX, 0000	BA0000	(AX) = 1A05 (DX) = 0000 (DS) = 1A05 (SP) = FFFC (IP) = 000E Stack +0 0000 +2 19F5 +4 6548	(AX) = 1A05 (DX) = 0000 (DS) = 1A05 (SP) = 0014 (IP) = 0011 Stack +0 0000 +2 19F5 +4 6548
0011	CALL 0000	E8ECFF	(AX) = 1A05 (DX) = 0000 (DS) = 1A05 (SP) = 0014 (IP) = 0011 Stack +0 0000 +2 19F5 +4 6548	(AX) = 1A05 (DX) = 0000 (DS) = 1A05 (SP) = FFFA (IP) = 0000 Stack +0 0014 +2 0000 +4 19F5
0000	MOV AH,09	B409	(AX) = 1A05 (DX) = 0000 (DS) = 1A05 (SP) = FFFA (IP) = 0000 Stack +0 0014 +2 0000 +4 19F5	(AX) = 0905 (DX) = 0000 (DS) = 1A05 (SP) = FFFA (IP) = 0002 Stack +0 0014 +2 0000 +4 19F5
0002	INT 21	CD21	(AX) = 0905 (DX) = 0000 (DS) = 1A05 (SP) = FFFA (IP) = 0002 Stack +0 0014 +2 0000 +4 19F5	(AX) = 0905 (DX) = 0000 (DS) = 1A05 (SP) = FFFA (IP) = 0004 Stack +0 0014 +2 0000 +4 19F5
0004	RET	C3	(AX) = 0905 (DX) = 0000 (DS) = 1A05 (SP) = FFFA (IP) = 0004 Stack +0 0014 +2 0000 +4 19F5	(AX) = 0905 (DX) = 0000 (DS) = 1A05 (SP) = FFFC (IP) = 0014 Stack +0 0000 +2 19F5 +4 6548
0014	MOV DX,0010	BA1000	(AX) = 0905 (DX) = 0000 (DS) = 1A05 (SP) = FFFC (IP) = 0014 Stack +0 0000 +2 19F5 +4 6548	(AX) = 0905 (DX) = 0010 (DS) = 1A05 (SP) = FFFC (IP) = 0014 Stack +0 0000 +2 19F5 +4 6548

0017	CALL 0000	E8E6FF	(AX) = 0905 (DX) = 0010 (DS) = 1A05 (SP) = FFFC (IP) = 0014 Stack +0 0000 +2 19F5 +4 6548	(AX) = 0905 (DX) = 000F (DS) = 1A05 (SP) = FFFA (IP) = 0000 Stack +0 001A +2 0000 +4 19F5
0000	MOV AH,09	B409	(AX) = 0905 (DX) = 000F (DS) = 1A05 (SP) = FFFA (IP) = 0000 Stack +0 001A +2 0000 +4 19F5	(AX) = 0905 (DX) = 000F (DS) = 1A05 (SP) = FFFA (IP) = 0002 Stack +0 001A +2 0000 +4 19F5
0002	INT 21	CD21	(AX) = 0905 (DX) = 000F (DS) = 1A05 (SP) = FFFA (IP) = 0002 Stack +0 001A +2 0000 +4 19F5	(AX) = 0905 (DX) = 000F (DS) = 1A05 (SP) = FFFA (IP) = 0004 Stack +0 001A +2 0000 +4 19F5
0004	RET	C3	(AX) = 0905 (DX) = 000F (DS) = 1A05 (SP) = FFFA (IP) = 0004 Stack +0 001A +2 0000 +4 19F5	(AX) = 0905 (DX) = 000F (DS) = 1A05 (SP) = FFFC (IP) = 001A Stack +0 0000 +2 19F5 +4 6548
001A	RET Far	CB	(AX) = 0905 (DX) = 000F (DS) = 1A05 (SP) = FFFC (IP) = 001A Stack +0 0000 +2 19F5 +4 6548	(AX) = 0905 (DX) = 000F (DS) = 1A05 (SP) = 0000 (IP) = 0000 Stack +0 6548 +2 6C6C +4 206F
0000	INT 20	CD20	(AX) = 0905 (DX) = 000F (DS) = 1A05 (SP) = 0000 (IP) = 0000 Stack +0 6548 +2 6C6C +4 206F	(AX) = 0905 (DX) = 000F (DS) = 1A05 (SP) = FFFA (IP) = 1480 Stack +0 0002 +2 19F5 +4 7244

Листинг:

```

= 0024                                EOFLine EQU '$' ;
ESBİГ BГB«BГB«BГBB
                                ЁВГ ГЎВЁВ¬ВЎВ®В«Г¬ВВ®В© ВЄВ®ВГЎГЎ ВГЎГ«
                                ; "E В®ВВГГ|
ГЎГ
                                ЎГ В®ВЄВЁ"

                                ; ВЎЎГЎГВГБЄ BİГ В®ВЈГ В¬В¬Г«

                                ;AStack SEGMENT STACK
                                DW 12 DUP(?) ;
0000 000C[
ESГЎВЎВ®В«ВЁГЎГЎГİ
                                12 ГЎВ«В®ВЎ Вİ В¬ГİГЎВЁ
                                ????
                                ]

                                ;AStack ENDS

                                ; ВЎЎ ВВГ«ВГ BİГ В®ВЈГ В¬В¬Г«

0000                                DATA SEGMENT

                                ; ВЎЎВЁГ ВГБЄГЎВЁВЎГ« В®ВİВЁГЎ ВВЁГİ В« ВВГ
                                «ГГ

0000 48 65 6C 6C 6F 20 HELLO DB 'Hello World!', 0AH,
0DH, EOFLine
                                57 6F 72 6C 64 21
                                0A 0D 24
000F 4C 69 68 61 73 68 GREETING DB 'Lihashva Anastasia from
9383$'
                                76 61 20 41 6E 61
                                73 74 61 73 69 61
                                20 66 72 6F 6D 20
                                39 33 38 33 24

```

002C DATA ENDS

; E B®B® BİΓ B®BJΓ B¬B¬Γ«

0000 CODE SEGMENT

ASSUME CS:Code DS:DATA SS:AStack

HELLO2.ASM(28): warning A4001: Extra characters on line

; BİΓ B®Γ|BİΓB®ΓJΓ BİBİΓS ΓŸBË ΓŸΓŸΓ B®B®BË

0000 WriteMsg PROC NEAR

0000 B4 09 mov AH,9

0002 CD 21 int 21h ; B™Γ«B\$B®BŸ
Γ®ΓJB®Γ|BË

BË DOS BİB® BİΓ BİΓ Γ«BŸ BBËΓ®

0004 C3 ret

0005 WriteMsg ENDP

; Ж' B®B«B®BŸB İİ BİΓ B®Γ|BİΓB®ΓJΓ

0005 Main PROC FAR

0005 1E push DS ; \ B™B®ΓİΓ BBİB-
BË

Bİ B®Γ BİΓŸ B ΓS B« PSP BŸ ΓŸΓŸBİB®Bİ

0006 2B C0 sub AX,AX ; > B®B«İİ
BİB®ΓŸB«BİB

®ΓJΓ®Γ®BİΓJB® BŸB®ΓŸΓŸΓŸ BB®BŸB«BİBBËİİ BİB®

0008 50 push AX ; / B®B®B¬ BB®Bİ
ret,

B\$ BŸBİΓ İË Γ®Γ®BİΓ® BİΓ B®Γ|BİΓB®ΓJΓ İJ.

0009 B8 ---- R mov AX,DATA ;
B™Ÿ BJΓ İ

JB\$B® ΓŸBİΓJB¬BİBİŸBB®JB®

000C 8E D8 mov DS,AX ;
İ BİBJBËİ

ŸΓŸΓ B® BBΓ«İİ.

000E BA 0000 R mov DX, OFFSET HELLO ;
B™Γ«BŸB®

B® B İB® B BİBİΓ BŸB®B®

```

0011 E8 0000 R          call WriteMsg          ;
ГЎГЎГ B®B

                                €BĚ BĪГ BĚBŷBĪГЎГЎГЎBŷBĚГĪ.

0014 BA 000F R          mov    DX, OFFSET GREETING ;
BЪЉГ«BŷB®

                                B¤ B  ГBĈГ  B BŷГŷB®Г B®B©

0017 E8 0000 R          call WriteMsg          ;
ГЎГЎГ B®B

                                €BĚ BĪГ BĚBŷBĪГЎГЎГЎBŷBĚГĪ.

001A CB                ret                      ;
BЪЉГ«ГГВ®

                                B¤ Bŷ DOS BĪB® BĈB®B¬ BB¤BĪ,

                                                                ; B
ГГВ®B¤

                                ГĪГ©BĪГ©ГŷГĪ Bŷ 1-B®B¬ ГŷB«B®BŷBĪ PSP.

001B                    Main          ENDP
001B                    CODE          ENDS

                                END Main

```

Segments and Groups:

Class	N a m e	Length	Align	Combine
CODE	001B	PARA	NONE
DATA	002C	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

51 Source Lines

51 Total Lines

12 Symbols

48008 + 459252 Bytes symbol space free

Выводы.

В результате выполнения лабораторной работы была освоена трансляция, изучено, как происходит выполнение и отладка программ на языке Ассемблера, а также разобраны структуры приведенных в работе программ.

Приложение.

Текст файла *HELLO1.asm*

```
DOSSEG ; Задание
сегментов под ДОС
.MODEL SMALL ; Модель па-
мяти-SMALL (Малая)
.STACK 100h ; Отвести под
Стек 256 байт
.DATA ; Начало сег-
мента данных
Greeting LABEL BYTE ; Текст при-
ветствия
DB 'Hello! Im Lihashva Anastasia from 9383',13,10,'$'
.CODE ; Начало сегмента
кода
mov ax, @data ; Загрузка в DS ад-
реса начала
mov ds, ax ; сегмента данных
mov dx, OFFSET Greeting ; Загрузка в dx
смещения
; адреса текста при-
ветствия
DisplayGreeting:
mov ah, 9 ; # функции ДОС пе-
чати строки
int 21h ; вывод на экран
приветствия
mov ah, 4ch ; # функции ДОС за-
вершения программы
int 21h ; завершение про-
граммы и выход в ДОС
END
```

Текст файла *HELLO2.asm*

```
EOFLine EQU '$'
AStack SEGMENT STACK
DW 12 DUP(?)
AStack ENDS
DATA SEGMENT
HELLO DB 'Hello World!', 0AH, 0DH, EOFLine
```

```

    GREETING    DB  'Lihashva  Anastasia  from  9383',  0AH,  0DH,
EOFLine
DATA          ENDS
CODE          SEGMENT
              ASSUME CS:Code DS:DATA SS:STACK
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

```