

**МИНОБРНАУКИ РОССИИ**  
**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: ТРАНСЛЯЦИИ, ОТЛАДКА И ВЫПОЛНЕНИЕ ПРОГРАММ НА ЯЗЫКЕ**  
**ASSEMBLER.**

Студент гр. 1303

Насонов Я.К.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

### **Цель работы.**

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

### **Задание.**

1. Просмотреть программу `hello1.asm`.

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

6. Выполнить программу в автоматическом режиме, убедиться в корректности ее работы и зафиксировать результат выполнения в протоколе.

7. Запустить выполнение программы под управлением отладчика.

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.

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. Просмотрена программа `hello1.asm`, которая формирует и выводит на экран приветствие пользователя с помощью функции ОС MSDOS, вызываемой через прерывание с номером 21H.

2. Разобрана структура и реализация каждого сегмента программы. Строка-приветствие преобразована в соответствии с личными данными.

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

4. Протранслирована программа с помощью строки

```
> masm hello1.asm
```

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

5. Скомпонован загрузочный модуль с помощью строки

```
> link hello1.obj
```

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

6. Выполнена программа в автоматическом режиме путем набора строки

```
> hello1.exe
```

7. Запущено выполнение программы под управлением отладчика с помощью команды

```
> afd hello1.exe
```

По той же последовательности действий протранслирована, скомпонована и выполнена программа файла `hello2.asm`.

Последовательность команд в файле hello1.exe:

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 (IP) = 0010	(AX) = 1A07 (DS) = 19F5 (IP) = 0013
0013	Mov DS, AX	8ED8	(AX) = 1A07 (DS) = 19F5 (IP) = 0013	(AX) = 1A07 (DS) = 1A07 (IP) = 0015
0015	Mov DX, 0000	BA0000	(AX) = 1A07 (DS) = 1A07 (IP) = 0015	(AX) = 1A07 (DS) = 1A07 (IP) = 0018
0018	Mov AH, 09	B409	(AX) = 1A07 (DS) = 1A07 (IP) = 0018	(AX) = 0907 (DS) = 1A07 (IP) = 001A
001A	Int 21	CD21	(AX) = 0907 (DS) = 1A07 (IP) = 001A	(AX) = 0907 (DS) = 1A07 (IP) = 001C
001C	Mov AH, 4C	B44C	(AX) = 0907 (DS) = 1A07 (IP) = 001C	(AX) = 4C07 (DS) = 1A07 (IP) = 001E
001E	Int 21	CD21	(AX) = 4C07 (DS) = 1A07 (IP) = 001E	(AX) = 0000 (DS) = 19F5 (IP) = 0010

Последовательность команд в файле hello2.exe:

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 (IP) = 0005 (SP) = 0000 Stack +0 0000	(AX) = 0000 (DS) = 19F5 (IP) = 0006 (SP) = 0016 Stack +0 19F5
0006	Sub AX, AX	2BC0	(AX) = 0000 (DS) = 19F5 (IP) = 0006	(AX) = 0000 (DS) = 19F5 (IP) = 0008
0008	Push AX	50	(AX) = 0000 (DS) = 19F5 (IP) = 0008 (SP) = 0016 Stack +0 19F5 +2 0000	(AX) = 0000 (DS) = 19F5 (IP) = 0009 (SP) = 0014 Stack +0 0000 +2 19F5
0009	Mov AX, 1A07	B8071A	(AX) = 0000 (DS) = 19F5 (IP) = 0009	(AX) = 1A07 (DS) = 19F5 (IP) = 000C
000C	Mov DS, AX	8ED8	(AX) = 1A07 (DS) = 19F5 (IP) = 000C	(AX) = 1A07 (DS) = 1A07 (IP) = 000E
000E	Mov DX, 0000	BA0000	(AX) = 1A07 (DS) = 1A07 (IP) = 000E	(AX) = 1A07 (DS) = 1A07 (IP) = 0011
0011	Call 0000	E8ECFF	(AX) = 1A07 (DS) = 1A07 (IP) = 0011 (SP) = 0014 Stack +0 0000 +2 19F5 +4 0000	(AX) = 1A07 (DS) = 1A07 (IP) = 0000 (SP) = 0012 Stack +0 0014 +2 0000 +4 19F5

0000	Mov AH, 09	B409	(AX) = 1A07 (DS) = 1A07 (IP) = 0000	(AX) = 0907 (DS) = 1A07 (IP) = 0002
0002	Int 21	CD21	(AX) = 0907 (DS) = 1A07 (IP) = 0002	(AX) = 0907 (DS) = 1A07 (IP) = 0004
0004	Ret	C3	(AX) = 0907 (DS) = 1A07 (IP) = 0004 (SP) = 0012 Stack +0 0014 +2 0000 +4 19F5	(AX) = 0907 (DS) = 1A07 (IP) = 0014 (SP) = 0014 Stack +0 0000 +2 19F5 +4 0000
0014	Mov DX, 0010	BA1000	(AX) = 0907 (DS) = 1A07 (IP) = 0014 (DX) = 0000	(AX) = 0907 (DS) = 1A07 (IP) = 0017 (DX) = 0010
0017	Call 0000	E8E6FF	(AX) = 0907 (DS) = 1A07 (IP) = 0017 (SP) = 0014 Stack +0 0000 +2 19F5 +4 0000	(AX) = 0907 (DS) = 1A07 (IP) = 0000 (SP) = 0012 Stack +0 001A +2 0000 +4 19F5
0000	Mov AH, 09	B409	(AX) = 0907 (DS) = 1A07 (IP) = 0000	(AX) = 0907 (DS) = 1A07 (IP) = 0002
0002	Int 21	CD21	(AX) = 0907 (DS) = 1A07 (IP) = 0002	(AX) = 0907 (DS) = 1A07 (IP) = 0004
0004	Ret	C3	(AX) = 0907 (DS) = 1A07 (IP) = 0004 (SP) = 0012 Stack +0 001A +2 0000 +4 19F5	(AX) = 0907 (DS) = 1A07 (IP) = 001A (SP) = 0014 Stack +0 0000 +2 19F5 +4 0000

001A	Ret Far	CB	(AX) = 0907 (DS) = 1A07 (IP) = 001A (SP) = 0014 Stack +0 0000 +2 19F5 +4 0000	(AX) = 0907 (DS) = 1A07 (IP) = 0000 (SP) = 0018 Stack +0 0000 +2 0000 +4 0000
0000	Int 20	CD20	(AX) = 0907 (DS) = 1A07 (IP) = 0000	(AX) = 0000 (DS) = 19F5 (IP) = 0005

### **Выводы**

В ходе создания отчёта, а именно записи последовательности команд были изучены основные принципы трансляции, отладки и выполнения программ на языке Assembler.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название 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 'Hello from student of group 1303 - Nasonov Yaroslav',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
```

Название файла: hello2.asm

```
EOFLine EQU '$' ; Определение символьной константы
; "Конец строки"

; Стек программы

ASSUME CS:CODE, SS:AStack ; указание сегмента - стек

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 1303 - Nasonov Yaroslav$'
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

```

Название файла: hello1.lst

Microsoft (R) Macro Assembler Version 5.10

9/22/22 13:44:18

Page 1-1

```

1
2      ; HELLO1.ASM - упрощенная в
      ерсия учебной програй
      ¼мы лаб.раб. N1
3      ;      по дисциплин
      е "Архитектура компью
      тера"
4      ; *****
      *****
5      ; Назначение: ПрограмИ
      ¼а формирует и выводит
      □ на экран приветствийИ
      μ
6      ;      пользователя

```

```

с помощью функции ДОС
"Вывод строки"
7      ;      (номер 09 преры
        вание 21h), которая:
8      ;      - обеспечивае
        т вывод на экран строИ
        °и символов,
9      ;      заканчивающ
        ейся знаком "$";
10     ;      - требует задИ
        °ния в регистре ah номе
        ра функции=09h,
11     ;      а в регистре
        dx - смещения адреса
        выводимой
12     ;      строки;
13     ;      - использует э
        □егистр ax и не сохраИ
        ½яет его
14     ;      содержимое.
15     ; *****
        *****
16
17     DOSSEG
        ; Задание сегмент
        ов под ДОС
18     .MODEL SMALL
        ; Модель памяти-SMA
        LL(Малая)
19     .STACK 100h

```

```

; Отвести под СтeИ
° 256 байт
20 .DATA
; Начало сегмента
данных
21 0000 Greeting LABEL BYTE
; Текст приветстИ
²ия
22 0000 48 65 6C 6C 6F 20 DB 'Hello from student of group 1303
- Nasonov Yaroslav',13,10,'$'

```

```
23    66 72 6F 6D 20 73
24    74 75 64 65 6E 74
25    20 6F 66 20 67 72
26    6F 75 70 20 31 33
27    30 33 20 2D 20 4E
28    61 73 6F 6E 6F 76
29    20 59 61 72 6F 73
30    6C 61 76 0D 0A 24
31                                     .CODE
                                     ; Начало сегмента код
                                     а
32 0000 B8 ---- R                mov ax, @data
                                     ; Загрузка в DS адреса
                                     начала
33 0003 8E D8                    mov ds, ax
                                     ; сегмента данных
34 0005 BA 0000 R                mov dx, OFFSET Greeting
                                     ; Загрузка в dx смещен
                                     ия
35
                                     ; адреса текста приве
                                     тствия
36 0008                        DisplayGreeting:
37 0008 B4 09                    mov ah, 9
                                     ; # функции ДОС печатИ
                                     , строки
```

38 000A CD 21	int 21h
	; вывод на экран прив
	етствия
39 000C B4 4C	mov ah, 4ch
	; # функции ДОС заверэ
	□ения программы
40 000E CD 21	int 21h
	; завершение програмЙ
	¼Ы и ВЫХОД в ДОС
41	END

## Symbols-1

## Segments and Groups:

N a m e	Length	Align	Combine	Class
DGROUP .....				GROUP
_DATA .....	0036	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 Total Lines

19 Symbols

47460 + 459800 Bytes symbol space free

0 Warning Errors

0 Severe Errors

Название файла: hello2.lst

Microsoft (R) Macro Assembler Version 5.10

9/22/22 13:47:22

Page 1-1

```
1          ; HELLO2 - Учебная програЙ
              ¼ма N2 лаб.раб.#1 по дисэ
              □иплине "Архитектура Й
              °омпьютера"
2          ;      Программа испоЙ
              »ьзует процедуру для Й
              ъечати строки
3          ;
4          ;      ТЕКСТ ПРОГРАММЫ
5
6 = 0024          EOFLine EQU '$'      ; ОпредЙ
              мление символьной кон
              станты
7          ;      "Кон
              ец строки"
8
```

```

9           ; Стек программы
10
11          ASSUME CS:CODE, SS:AStack
12
13 0000          AStack  SEGMENT STACK
14 0000 000C[          DW 12 DUP('!') ; ОтвоЙ
                        'ится 12 слов памяти
15 0021
16           ]
17
18 0018          AStack  ENDS
19
20           ; Данные программы
21
22 0000          DATA   SEGMENT
23
24           ; Директивы описания
                        данных
25
26 0000 48 65 6C 6C 6F 20      HELLO  DB 'Hello Worlds!', 0AH, 0DH,
                        EOFLine
27 57 6F 72 6C 64 73
28 21 0A 0D 24
29 0010 53 74 75 64 65 6E GREETING DB 'Student from 1303 - Nason
                        ov Yaroslav$'
30 74 20 66 72 6F 6D
31 20 31 33 30 33 20
32 2D 20 4E 61 73 6F
33 6E 6F 76 20 59 61
34 72 6F 73 6C 61 76

```



35 24

36 0035 DATA ENDS

37

38 ; Код программы

39

40 0000 CODE SEGMENT

41 ; Процедура печати стр

□оки

```
42 0000      WriteMsg PROC NEAR
43 0000 B4 09      mov  AH,9
44 0002 CD 21      int  21h ; Вызов функции DOS по прерыванию
                                1/2
45 0004 C3      ret
46 0005      WriteMsg ENDP
47
48      ; Главная процедура
49 0005      Main  PROC FAR
50 0005 1E      push DS      ; \ Сохранение адреса начала PS
                                Р в стеке
51 0006 2B C0      sub  AX,AX  ; > для п1
                                3/4
                                следующего восстановления
                                2
52 0008 50      push AX      ; / команда ret, завершающей процедуру.
53 0009 B8 ---- R      mov  AX,DATA      ; Инициализация
                                1
                                загрузки сегментного регистра
54 000C 8E D8      mov  DS,AX      ; Э
                                2
                                регистра данных.
55 000E BA 0000 R      mov  DX, OFFSET HELLO ; Инициализация
                                3
                                вывода на экран первой строки
56 0011 E8 0000 R      call WriteMsg      ; Э
                                4
                                строки приветствия.
```

57 0014 BA 0010 R	mov DX, OFFSET GREETING ; Й
	□ывод на экран второй
58 0017 E8 0000 R	call WriteMsg ; э
	□троки приветствия.
59 001A CB	ret ; Й
	□ыход в DOS по команде,
60	; Й
	½аходящейся в 1-ом слоЙ
	²е PSP.
61 001B	Main ENDP
62 001B	CODE ENDS
63	END Main

## Symbols-1

## Segments and Groups:

N a m e	Length	Align	Combine	Class
ASTACK .....	0018	PARA		STACK
CODE .....	001B	PARA		NONE
DATA .....	0035	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	CODE      Length = 0016
WRITEMSG .....	N PROC	0000	CODE      Length = 0005
@CPU .....	TEXT	0101h	
@FILENAME .....	TEXT	hello2	
@VERSION .....	TEXT	510	

52 Source Lines

52 Total Lines

13 Symbols

47452 + 459805 Bytes symbol space free

0 Warning Errors

0 Severe Errors