

**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: Трансляция, отладка и выполнение программ на языке**  
**Ассемблера**

Студент гр. 0383

\_\_\_\_\_

Куртова К. А.

Преподаватель

\_\_\_\_\_

Ефремов М. А.

Санкт-Петербург

2021

## Цель работы.

Изучить работу программ hello1.asm и hello2.asm, выводящих на экран текст сообщения, с помощью отладчика. Представить протокол с фиксацией всех выполняемых действий и полученных результатов. Сравнить результаты прогона программ в отладчике и объяснить различия в программах.

## Ход работы.

1. Каталог с MASM и программами смонтирован в эмулятор командой  
`mount d d:/tools`
2. Программа hello1.asm (см. приложение А) протранслирована с помощью MASM, созданы объектный (hello1.obj) и листинговый (hello1.lst — см. приложение Б) файлы. Синтаксические ошибки не были обнаружены транслятором (см. рис. 1).

```
47994 + 461313 Bytes symbol space free

0 Warning Errors
0 Severe Errors

D:\>dir
Содержимое D:\.
.           <DIR>           06-09-2021 17:47
..          <DIR>           01-01-1980  0:00
AFDPRO  EXE           50,937 25-06-2006 23:14
HELLO1  ASM            1,899 06-09-2021 12:55
HELLO1  LST            3,916 06-09-2021 17:47
HELLO1  OBJ             203 06-09-2021 17:47
HELLO2  ASM            2,063 02-09-2021 20:02
LINK    EXE           65,475 25-06-2006 23:14
LR2     DAT            4,406 02-09-2021 20:01
MASM    EXE          110,703 26-06-2006  0:14
      8 Файл(ов)           239,602 Байт.
      2 Директорий         262,111,744 Байт свободно.
```

Рисунок 1 — Трансляция программы и её результат

3. Загрузочный модуль скомпонован с помощью команды  
`link hello1.obj,`  
в результате получены исполняемый файл (hello1.exe) и карта памяти (hello1.map).
4. Запущена программа hello1.exe. Выводимое сообщение было преобразовано, исходный код был сохранён в кодировке OEM (866)

для корректного отображения сообщения. Результат работы программы представлен на рис. 2.

```
D:\>hello1.exe
Вас приветствует ст.гр. 0383 – Куртова К. А.
D:\>
```

Рисунок 2 — Результат выполнения программы hello1.exe

5. Программа запущена в отладчике командой

afopro.exe hello1.exe

6. Начальное содержание сегментных регистров для программы hello1.exe:

(CS) = 1A05

(DS) = 19F5

(ES) = 19F5

(SS) = 1A0A

Результат прогона программы под управлением отладчика представлен в табл. 1.

Таблица 1 — протокол выполнения программы hello1.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	B8071A	(AX) = 0000 (IP) = 0010	(AX) = 1A07 (IP) = 0013
0013	Mov DS, AX	8ED8	(DS) = 0000 (IP) = 0013	(DS) = 1A07 (IP) = 0015
0015	Mov DX, 0000	BA0000	(IP) = 0015	(IP) = 0018
0018	Mov AH, 09	B409	(AX) = 1A07 (IP) = 0018	(AX) = 0907 (IP) = 001A

Продолжение таблицы 1

001A	Int 21	CD21	(IP) = 001A	(IP) = 001C
001C	Mov AH, C4	B44C	(IP) = 001C (AX) = 0917	(IP) = 001E (AX) = 4C07
001E	Int 21	CD21	(IP) = 001E	(IP) = 0010

7. Для hello2.asm (см. приложение В) аналогично создаются объектный, листинговый (см. приложение Г), исполняемый файл и карта памяти. Результат работы программы представлен на рис. 3.

```
D:\>hello2.exe
Hello Worlds!
Student from 0383 - K. K.
D:\>_
```

Рисунок 3 — Результат работы программы hello2.exe

8. Начальное содержание сегментных регистров для программы hello1.exe:

(CS) = 1A0A

(DS) = 19F5

(ES) = 19F5

(SS) = 1A05

Результат прогона программы под управлением отладчика представлен в табл. 2.

Таблица 2 — протокол выполнения 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	(IP) = 0006
0006	Sub AX, AX	2BC0	(IP) = 0006	(IP) = 0008
0008	Push AX	50	(IP) = 0008	(IP) = 0009
0009	Mov AX, 1A07	B8071A	(IP) = 0009 (AX) = 0000	(IP) = 000C (AX) = 1A07
000C	Mov DS, AX	8ED8	(IP) = 000C (DS) = 19F5	(IP) = 000E (DS) = 1A07
000E	Mov DX, 0000	BA0000	(IP) = 000E	(IP) = 0011
0011	Call 0000	E8ECFF	(IP) = 0011	(IP) = 0000
0000	Mov AH, 09	B409	(IP) = 0000 (AX) = 1A07	(IP) = 0002 (AX) = 0907
0002	Int 21	CD21	(IP) = 0002	(IP) = 0004
0004	Ret	C3	(IP) = 0004	(IP) = 0014
0014	Mov DX, 0010	BA1000	(IP) = 0014 (DX) = 0000	(IP) = 0017 (DX) = 0010
0017	Call 0000	E8E6FF	(IP) = 0017	(IP) = 0000
0000	Mov AX, 09	B409	(IP) = 0000	(IP) = 0002
0002	Int 21	CD21	(IP) = 0002	(IP) = 0004
0004	Ret	C3	(IP) = 0004	(IP) = 001A
001A	Ret	CB	(IP) = 001A (CS) = 1A0A	(IP) = 0000 (CS) = 19F5
0000	Int 20	CD20	(IP) = 0000	(IP) = 0005

9. Комментарии по смыслу исполняемых команд.

hello1.asm:

- 1) Определяется метка Greetings типа BYTE.
- 2) В сегменте кода (CS): в регистр AX помещается смещение сегмента, в котором находятся данные (@data), т. к. нельзя напрямую изменять содержимое регистра DS. Команда mov ds, ax устанавливает в DS значение смещения, хранящееся в AX.
- 3) В регистр DX помещается значение смещения начала сообщения (OFFSET). В данном случае смещение равно 0000.
- 4) Метка DisplayGreetings: mov ah, 9 соответствует функции вывода сообщения, прерывание int 21h выполняет функцию, номер которой был передан в AH.
- 5) Аналогично, mov ah, 4ch соответствует функции завершения программы.

hello2.asm:

- 1) ASSUME сообщает ассемблеру, какой регистр сегмента используется для доступа к информации. В данном случае используются сегмент кода (CS) и сегмент стека (SS). Для сегмента стека выделяется память.
- 2) Сегмент данных: описываются сообщения, которые необходимо вывести (HELLO, GREETING).
- 3) Сегмент кода: определяется процедура вывода сообщения аналогично выводу сообщения в программе hello1.asm.
- 4) В стек сохраняются значения DS и AX.
- 5) В регистр DX помещается значение смещения, находящееся в AX, аналогично тому, как это сделано в hello1.asm.
- 6) В DX помещается значение смещения сообщения HELLO, после чего вызывается процедура WriteMsg с помощью команды call. Call записывает в стек адрес команды, команда ret используется

для возвращения к вызывающей процедуре, что уменьшает указатель вершины стека.

7) Аналогично вызывается сообщение GREETING.

8) Программа завершается.

### **Выводы.**

В ходе лабораторной работы ознакомились с эмулятором DOSBox, научились работать с отладчиком AFD Pro и рассмотрели простейшие программы на языке ассемблера. Был рассмотрен базовый синтаксис языка и изучены разные подходы к реализации задания, в данном случае — печати строки.

## ПРИЛОЖЕНИЕ А

### Текст исходного файла программы hello1.exe

; 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. 0383 - Куртова К. А.',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

```

## ПРИЛОЖЕНИЕ Б

### Текст файла диагностических сообщений программы **hello1.lst**

Microsoft (R) Macro Assembler Version 5.10

9/6/21 17:47:07

Page 1-1

; 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 ДО  
С "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 регистры 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

0000 Greeting LABEL BYTE

; Текст приветствия

0000 82 A0 E1 20 AF E0 DB 'Вас приветствует ст.гр. 0383 - Куртова К  
. A.',13,10,'\$'

A8 A2 A5 E2 E1 E2

A2 E3 A5 E2 20 E1

E2 2E A3 E0 2E 20

30 33 38 33 20 2D

20 8A E3 E0 E2 AE

A2 A0 20 8A 2E 20

80 2E 0D 0A 24

.CODE ; Начал

о сегмента кода

0000 B8 ---- R mov ax, @data ; Загр

узка в DS адреса начала

0003 8E D8 mov ds, ax ; сегм

ента данных

0005 BA 0000 R mov dx, OFFSET Greeting ; Загр

узка в dx смещения

Microsoft (R) Macro Assembler Version 5.10 9/6/21 17:47:07

Page 1-2

; адрес

а текста приветствия

0008 DisplayGreeting:

0008 B4 09 mov ah, 9 ; # фу

нкции ДООС печати строки

000A CD 21 int 21h ; ВЫВОД

на экран приветствия

000C B4 4C mov ah, 4ch ; # фу

нкции ДООС завершения программы

000E CD 21 int 21h ; завер

шение программы и выход в ДООС

END

Microsoft (R) Macro Assembler Version 5.10 9/6/21 17:47:07

Symbols-1

Segments and Groups:

N a m e	Length	Align	Combine	Class
DGROUP .....	GROUP			
_DATA .....	002F 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

47994 + 461313 Bytes symbol space free

0 Warning Errors

0 Severe Errors

## ПРИЛОЖЕНИЕ В

### Текст исходного файла программы hello2.exe

; 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 4350 - \$'

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 Г

### Текст файла диагностических сообщений программы **hello2.lst**

Microsoft (R) Macro Assembler Version 5.10

9/6/21 21:07:12

Page 1-1

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

; Программа использует процедуру для п  
ечати строки

;

; ТЕКСТ ПРОГРАММЫ

= 0024 EOFLine EQU '\$' ; Определение символъ  
ной константы  
; "Конец строки"

; Стек программы

ASSUME CS:CODE, SS:AStack

0000 AStack SEGMENT STACK

0000 000C[ DW 12 DUP('') ; Отводится 12 слов  
памяти

0021

]

0018 AStack ENDS



; Данные программы

0000 DATA SEGMENT

; Директивы описания данных

0000 48 65 6C 6C 6F 20 HELLO DB 'Hello Worlds!', 0AH,  
0DH,EOFLine

57 6F 72 6C 64 73

21 0A 0D 24

0010 53 74 75 64 65 6E GREETING DB 'Student from 0383 - K. K.\$'

74 20 66 72 6F 6D

20 30 33 38 33 20

2D 20 4B 2E 20 4B

2E 24

002A DATA ENDS

; Код программы

0000 CODE SEGMENT

; Процедура печати строки

0000 WriteMsg PROC NEAR

0000 B4 09 mov AH,9

0002 CD 21 int 21h ; Вызов функции DOS по пре  
рыванию

0004 C3 ret

0005 WriteMsg ENDP

; Головная процедура

0005 Main PROC FAR

0005	1E	push DS	; \ Сохранение адреса
		начала PSP в стеке	
Microsoft (R) Macro Assembler Version 5.10		9/6/21 21:07:12	
Page 1-2			
0006	2B C0	sub AX,AX	; > для последующего в
		остановления по	
0008	50	push AX	; / команде ret, завер
		шающей процедуру.	
0009	B8 ---- R	mov AX,DATA	; Загрузка
		сегментного	
000C	8E D8	mov DS,AX	; регистра
		данных.	
000E	BA 0000 R	mov DX, OFFSET HELLO	; Вывод на
		экран первой	
0011	E8 0000 R	call WriteMsg	; строки пр
		иветствия.	
0014	BA 0010 R	mov DX, OFFSET GREETING	; Вывод
на			
		экран второй	
0017	E8 0000 R	call WriteMsg	; строки пр
		иветствия.	
001A	CB	ret	; Выход в D
		OS по команде,	
		; находящей	
		ся в 1-ом слове PSP.	
001B		Main	ENDP
001B		CODE	ENDS
		END Main	

## Symbols-1

## Segments and Groups:

N a m e	Length	Align	Combine	Class
ASTACK .....	0018	PARA		STACK
CODE .....	001B	PARA		NONE
DATA .....	002A	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

47986 + 459274 Bytes symbol space free

0 Warning Errors

0 Severe Errors