

**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. 1381

\_\_\_\_\_

Смирнов Д. Ю.

Преподаватель

\_\_\_\_\_

Ефремов М.А.

Санкт-Петербург

2022

## Цель работы.

Изучить работу простейших программ на языке Ассемблера, выполнить их трансляцию, отладку и компоновку.

## Задание.

### Часть 1

1. Просмотреть программу *hello1.asm*, которая формирует и выводит на экран приветствие пользователя с помощью функции *OC 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. Непонятные фрагменты прояснить у преподавателя. 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 транслятором. Повторить трансляцию программы до получения объектного модуля.

5. Скомпоновать загрузочный модуль с помощью строки

*> link hello1.obj*

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

6. Выполнить программу в автоматическом режиме путем набора строки

*> hello1.exe*

убедиться в корректности ее работы и зафиксировать результат выполнения в протоколе.

7. Запустить выполнение программы под управлением отладчика с помощью команды

*> afd hello1.exe*

#### Часть 2

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.**

- 1.1. Прочитан код программы *HELLO1.ASM*, строка с приветствием была отредактирована в соответствии с личными данными.
- 1.2. В DOSBox была загружена кодовая страница с буквами кириллицы, командой *keyb ru 866*.
- 1.3. Выполнена трансляция *HELLO1.ASM* в *HELLO1.OBJ* (командой *MASM.EXE HELLO1.ASM*), был создан файл листинга *HELLO1.LST*. Ошибок при трансляции не обнаружено.
- 1.4. Компоновка загрузочного модуля объектного файла *HELLO1.OBJ*, создан исполняемый файл *HELLO1.EXE* с созданием карты памяти *HELLO1.MAP*.
- 1.5. Запуск исполняемого файла *HELLO1.EXE* в автоматическом режиме, в консоль была выведена строка «Вас приветствует ст.гр.1381 - Смирнов Д.Ю.».
- 1.6. Используя отладчик *AFDPRO*, был пошагово отсмотрен исполняемый файл программы *HELLO1.EXE*, результат прогона см. Таблица 1.

- 2.1. Прочитан код программы *HELLO2.ASM*, строка с приветствием была отредактирована в соответствии с личными данными.
- 2.2. Выполнена трансляция *HELLO2.ASM* в *HELLO2.OBJ* (командой *MASM.EXE HELLO2.ASM*), был создан файл листинга *HELLO2.LST*. Ошибок при трансляции не обнаружено.
- 2.3. Компоновка загрузочного модуля объектного файла *HELLO2.OBJ*, создан исполняемый файл *HELLO2.EXE* с созданием карты памяти *HELLO2.MAP*.
- 2.4. Запуск исполняемого файла *HELLO2.EXE* в автоматическом режиме, в консоль была выведена строка «Hello Worlds! Student from 1381 - Smirnov Dmitriy».
- 2.5. Используя отладчик *AFDPRO*, был пошагово отсмотрен исполняемый файл программы *HELLO2.EXE*, результат прогона см. Таблица 2.

Программный код *HELLO1.ASM* и файл листинга *HELLO1.LST* см. в приложении А.

Программный код *HELLO2.ASM* и файл листинга *HELLO2.LST* см. в приложении Б.

Таблица 1 - Результаты прогона программы *HELLO1.EXE* в отладчике  
Начальные значения: (CS)=1A05, (DS)=19F5, (ES)=19F5, (SS)=1A0A

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 (DX)=0000 (IP)=0015	(AX)=1A07 (DS)=1A07 (DX)=0000 (IP)=0018
0018	MOV AH, 9	B409	(AX)=1A07 (DS)=1A07 (DX)=0000 (IP)=0018	(AX)=0907 (DS)=1A07 (DX)=0000 (IP)=001A
001A	INT 21h	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 21h	CD21	(AX)=4C07 (DS)=1A07 (DX)=0000 (IP)=001E	Программа завершена

Таблица 2 - Результаты прогона программы *HELLO2.EXE* в отладчике  
Начальные значения: (CS)=1A0A, (DS)=19F5, (ES)=19F5, (SS)=1A05

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0005	PUSH DS	1E	(AX)=0000 (DX)=0000 (DS)=19F5 (SS)=1A05 (SP)=0018 (IP)=0005 Stack= +0 0000	(AX)=0000 (DX)=0000 (DS)=19F5 (SS)=1A05 (SP)=0016 (IP)=0006 Stack= +0 19F5
0006	SUB AX, AX	2BC0	(AX)=0000 (DX)=0000 (DS)=19F5 (SS)=1A05 (SP)=0016 (IP)=0006 Stack= +0 19F5	(AX)=0000 (DX)=0000 (DS)=19F5 (SS)=1A05 (SP)=0016 (IP)=0008 Stack= +0 19F5
0008	PUSH AX	50	(AX)=0000 (DX)=0000 (DS)=19F5 (SS)=1A05	(AX)=0000 (DX)=0000 (DS)=19F5 (SS)=1A05

			(SP) = 0016 (IP) = 0008 Stack = +0 19F5	(SP)=0014 (IP)=0009 Stack= +0 0000 +2 19F5
0009	MOV AX, 1A07	B8071A	(AX)=0000 (DX)=0000 (DS)=19F5 (SS)=1A05 (SP)=0014 (IP)=0009 Stack= +0 0000 +2 19F5	(AX)=1A07 (DX)=0000 (DS)=19F5 (SS)=1A05 (SP)=0014 (IP)=000C Stack= +0 0000 +2 19F5
000C	MOV DS, AX	8ED8	(AX)=1A07 (DX)=0000 (DS)=19F5 (SS)=1A05 (SP)=0014 (IP)=000C Stack= +0 0000 +2 19F5	(AX)=1A07 (DX)=0000 (DS)=1A07 (SS)=1A05 (SP)=0014 (IP)=000E Stack= +0 0000 +2 19F5
000E	MOV DX, 0000	BA0000	(AX)=1A07 (DX)=0000 (DS)=1A07 (SS)=1A05 (SP)=0014 (IP)=000C Stack= +0 0000 +2 19F5	(AX)=1A07 (DX)=0000 (DS)=1A07 (SS)=1A05 (SP)=0014 (IP)=0011 Stack= +0 0000 +2 19F5
0011	CALL 0000	E8CFF	(AX)=1A07 (DX)=0000 (DS)=1A07 (SS)=1A05 (SP)=0014 (IP)=0011 Stack= +0 0000 +2 19F5	(AX)=1A07 (DX)=0000 (DS)=1A07 (SS)=1A05 (SP)=0012 (IP)=0000 Stack= +0 0014 +2 0000 +4 19F5
0000	MOV AH, 09	B409	(AX)=1A07 (DX)=0000 (DS)=1A07 (SS)=1A05 (SP)=0012 (IP)=0000 Stack= +0 0014 +2 0000 +4 19F5	(AX)=0907 (DX)=0000 (DS)=1A07 (SS)=1A05 (SP)=0012 (IP)=0002 Stack= +0 0014 +2 0000 +4 19F5

0002	INT 21	CD21	(AX)=0907 (DX)=0000 (DS)=1A07 (SS)=1A05 (SP)=0012 (IP)=0002 Stack= +0 0014 +2 0000 +4 19F5	(AX)=0907 (DX)= 0000 (DS)=1A07 (SS)=1A05 (SP)=0012 (IP)=0004 Stack= +0 0014 +2 0000 +4 19F5
0004	RET	C3	(AX)=0907 (DX)=0000 (DS)=1A07 (SS)=1A05 (SP)=0012 (IP)=0004 Stack= +0 0014 +2 0000 +4 19F5	(AX)=0907 (DX)=0000 (DS)=1A07 (SS)=1A05 (SP)=0014 (IP)=0014 Stack= +0 0000 +2 19F5
0014	MOV DX, 0010	BA1000	(AX)=0907 (DX)=0000 (DS)=1A07 (SS)=1A05 (SP)=0014 (IP)=0014 Stack= +0 0000 +2 19F5	(AX)=0907 (DX)=0010 (DS)=1A07 (SS)=1A05 (SP)=0014 (IP)=0017 Stack= +0 0000 +2 19F5
0017	CALL 0000	E8E6FF	(AX)=0907 (DX)=0010 (DS)=1A07 (SS)=1A05 (SP)=0014 (IP)=0017 Stack= +0 0000 +2 19F5	(AX) =0907 (DX) = 0010 (DS)=1A07 (SS)=1A05 (SP)=0012 (IP)=0000 Stack= +0 001A +2 0000 +4 19F5
0000	MOV AX, 09	B409	(AX)=0907 (DX)=0010 (DS)=1A07 (SS)=1A05 (SP)=0012 (IP)=0000 Stack= +0 001A +2 0000 +4 19F5	(AX)=0907 (DX)=0010 (DS)=1A07 (SS)=1A05 (SP)=0012 (IP)=0002 Stack= +0 001A +2 0000 +4 19F5
0002	INT 21	CD21	(AX)=0907 (DX)=0010 (DS)=1A07	(AX)=0907 (DX)=0010 (DS)=1A07

			(SS)=1A05 (SP)=0012 (IP)=0002 Stack= +0 001A +2 0000 +4 19F5	(SS)=1A05 (SP)=0012 (IP)=0004 Stack= +0 001A +2 0000 +4 19F5
0004	RET	C3	(AX)=0907 (DX)=0010 (DS)=1A07 (SS)=1A05 (SP)=0012 (IP)=0004 Stack= +0 001A +2 0000 +4 19F5	(AX)=0907 (DX)=0010 (DS)=1A07 (SS)=1A05 (SP)=0014 (IP)=001A Stack= +0 0000 +2 19F5
001A	RET Far	CB	(AX)=0907 (DX)=0010 (DS)=1A07 (SS)=1A05 (SP)=0014 (CS)=1A0A (IP)=001A Stack= +0 0000 +2 19F5	(AX)=0907 (DX)=0010 (DS)=1A07 (SS)=1A05 (SP)=0018 (CS)=19F5 (IP)=0000 Stack= +0 0000
0000	INT 20	CD20	(AX)=0907 (DX)=0010 (DS)=1A07 (SS)=1A05 (SP)=0018 (CS)=19F5 (IP)=0000 Stack= +0 0000	Программа завершена



### **Выводы.**

В ходе лабораторной работы были изучены процессы трансляции, отладки и выполнения программ на языке Ассемблера.

## ПРИЛОЖЕНИЕ А

### Название файла: HELLO1.ASM

```
; 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.1381 - Смирнов Д.Ю.',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: HELLO1.LST

Microsoft (R) Macro Assembler Version 5.10

9/11/22 19:16:53

Page 1-1

```
DOSSEG
.MODEL SMALL
.STACK 100h
.DATA
0000 Greeting LABEL BYTE
0000 82 A0 E1 20 AF E0 DB 'Вас приветствует ст.гр.1381 - Смирнов
Д.Ю.',13,10,'$'
      A8 A2 A5 E2 E1 E2
      A2 E3 A5 E2 20 E1
      E2 2E A3 E0 2E 31
```

```

33 38 31 20 2D 20
91 AC A8 E0 AD AE
A2 20 84 2E 9E 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

```

Microsoft (R) Macro Assembler Version 5.10

9/11/22 19:16:53

Page 1-2

```

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

9/11/22 19:16:53

Symbols-1

# Segments and Groups:

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

```

0 Warning Errors
0 Severe Errors

```

## ПРИЛОЖЕНИЕ Б

Название файла: HELLO2.ASM

; 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 1381 - Smirnov Dmitriy \$'

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: HELLO2.LST

Microsoft (R) Macro Assembler Version 5.10

9/11/22 19:19:08

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 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 1381 - Smirnov
Dmitriy $'
              74 20 66 72 6F 6D
              20 31 33 38 31 20
              2D 20 53 6D 69 72
              6E 6F 76 20 44 6D
              69 74 72 69 79 20
              24

0035          DATA     ENDS
0000          CODE      SEGMENT
0000          WriteMsg  PROC    NEAR
0000 B4 09                      mov     AH,9
0002 CD 21                      int     21h
Microsoft (R) Macro Assembler Version 5.10

```

9/11/22 19:19:08

Page 1-2

```

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 0010 R                mov     DX, OFFSET GREETING
0017 E8 0000 R                call    WriteMsg
001A CB                      ret
001B          Main      ENDP
001B          CODE      ENDS
                      END Main

```

## 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

47986 + 459271 Bytes symbol space free

0 Warning Errors  
 0 Severe Errors