

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МОЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №1**  
**по дисциплине «Организация ЭВМ и систем»**  
**Тема: Освоение трансляции, выполнения и отладки программ на языке**  
**Ассемблера**

Студентка гр.1381

\_\_\_\_\_

Рымарь М.И.

Преподаватель

\_\_\_\_\_

Ефремов М.А.

Санкт-Петербург

2022

## **Цель работы.**

Познакомиться с работой простейших программ на языке Ассемблер. Выполнить их трансляцию, компоновку и отладку.

## **Задание.**

### *Часть 1*

1. Просмотреть программу `hello1.asm`, которая формирует и выводит на экран приветствие пользователя с помощью функции ОС 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 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 и 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.

## 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.

### Часть 1:

1. Просмотрена и разобрана структура программы *hello1.asm*, строка приветствия изменена в соответствии с личными данными.
2. Программа протранслирована с созданием объектного файла *hello1.obj* и листингового файла *hello1.lst*.
3. Скомпонован загрузочный модуль с созданием карты памяти *hello1.map* и исполняемого файла *hello1.exe*.
4. Программа запущена, строка приветствия выведена корректно.
5. Посредством отладчика была запущена программа. Начальные значения регистров, включая сегментные: (AX) = 0000, (BX) = 0000, (CX) = 004E, (DX) = 0000, (CS) = 11AC, (DS) = 119C, (ES) = 119C, (SS) = 11B1. Результаты пошагового выполнения представлены в таблице 1.

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров	
			До выполнения	После выполнения
0010	MOV AX, 11AE	B8AE11	(AX) = 0000 (IP) = 0010	(AX) = 11AE (IP) = 0013

0013	MOV DS, AX	8ED8	(DS) = 119C (IP) = 0013 (AX) = 11AE	(DS) = 11AE (IP) = 0015 (AX) = 11AE
0015	MOV DX, 0000	BA0000	(IP) = 0015 (DX) = 0000	(IP) = 0018 (DX) = 0000
0018	MOV AH, 09	B409	(AX) = 11AE (IP) = 0018	(AX) = 09AE (IP) = 001A
001A	INT 21	CD21	(IP) = 001A	(IP) = 001C
001C	MOV AH, 4C	B44C	(AX) = 09AE (IP) = 001C	(AX) = 4CAE (IP) = 001E
001E	INT 21	CD21	(AX) = 4CAE (CX) = 004E (DS) = 11AE (IP) = 001E	(AX) = 0000 (CX) = 0000 (DS) = 119C (IP) = 0010

Таблица 1 – Пошаговое выполнение программы *hello1.exe*

## Часть 2:

1. Для файла *hello2.asm* были проделаны все те же шаги, что и для предыдущего: просмотрена и разобрана программа, изменена строка приветствия, программа протранслирована с созданием объектного и листингового файлов, скомпонован загрузочный файл с созданием карты памяти и исполняемого файла. Программа запущена в автоматическом режиме, строка приветствия выводится корректно. С помощью отладчика запущено выполнение программы. Начальные значения регистров, включая сегментные: (AX) = 0000, (BX) = 0000, (CX) = 007B, (DX) = 0000, (CS) = 11B2, (DS) = 119C, (ES) = 119C, (SS) = 11AC. Результаты пошагового выполнения представлены в таблице 2.

2. Проведён анализ результатов прогона программ *hello1* и *hello2*. Обнаружены различия в размещении сегментов. В первой программе используется упрощённое определение сегментов (команда DOSSEG), а во второй – полное.

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 (DS) = 119C Stack: +0 0000	(SP) = 0016 (IP) = 0006 (DS) = 119C Stack: +0 119C
0006	SUB AX, AX	2BC0	(AX) = 0000 (IP) = 0006	(AX) = 0000 (IP) = 0008
0008	PUSH AX	50	(SP) = 0016 (IP) = 0008 (AX) = 0000 Stack: +0 119C +2 0000	(SP) = 0014 (IP) = 0009 (AX) = 0000 Stack: +0 0000 +2 119C
0009	MOV AX, 11AE	B8AE11	(AX) = 0000 (IP) = 0009	(AX) = 11AE (IP) = 000C
000C	MOV DS, AX	8ED8	(DS) = 119C (IP) = 000C (AX) = 11AE	(DS) = 11AE (IP) = 000E (AX) = 11AE
000E	MOV DX, 0000	BA0000	(IP) = 000E (DX) =	(IP) = 0011 (DX) =
0011	CALL 0000	E8ECFF	(SP) = 0014 (IP) = 0011 Stack: +0 0000 +2 119C +4 0000	(SP) = 0012 (IP) = 0000 Stack: +0 0014 +2 0000 +4 119C
0000	MOV AH, 09	B409	(AX) = 11AE (IP) = 0000	(AX) = 09AE (IP) = 0002
0002	INT 21	CD21	(IP) = 0002	(IP) = 0004
0004	RET	C3	(SP) = 0012 (IP) = 0004 Stack: +0 0014 +2 0000	(SP) = 0014 (IP) = 0014 Stack: +0 0000 +2 119C

			+4 119C	+4 0000
0014	MOV DX, 0010	BA1000	(IP) = 0014 (DX) = 0000	(IP) = 0017 (DX) = 0000
0017	CALL 0000	E8E6FF	(SP) = 0014 (IP) = 0017 Stack: +0 0000 +2 119C +4 0000	(SP) = 0012 (IP) = 0000 Stack: +0 001A +2 0000 +4 119C
0000	MOV AH, 09	B409	(IP) = 0000	(IP) = 0002
0002	INT 21	CD21	(IP) = 0002	(IP) = 0004
0004	RET	C3	(SP) = 0012 (IP) = 0004 Stack: +0 001A +2 0000 +4 119C	(SP) = 0014 (IP) = 001A Stack: +0 0000 +2 119C +4 0000
001A	RET Far	CB	(SP) = 0014 (CS) = 11B2 (IP) = 001A Stack: +2 119C	(SP) = 0018 (CS) = 119C (IP) = 0000 Stack: +2 0000
0000	INT 20	CD20	(AX) = 09AE (CX) = 007B (CS) = 119C (DS) = 11AE (IP) = 0000	(AX) = 0000 (CX) = 0000 (CS) = 11B2 (DS) = 119C (IP) = 0005

Таблица 2 – Пошаговое выполнение программы hello2.exe

### Выводы.

В ходе выполнения лабораторной работы был изучен синтаксис Ассемблера, рассмотрены процессы трансляции, компоновки и отладки, а также методы работы с сегментной памятью.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММ

Название файла: *hello1.asm*

```
DOSSEG
.MODEL    SMALL
.STACK    100H
.DATA
GREETING LABEL BYTE
    DB 'ВАС ПРИВЕТСТВУЕТ СТ.ГР.1381 - РЫМАРЬ МАРИЯ.',13,10,'$'
.CODE
MOV  AX,@DATA
MOV  DS,AX
MOV  DX,OFFSET GREETING
DISPLAYGREETING:
    MOV  AH,9
    INT  21H
    MOV  AH,4CH
    INT  21H
END
```

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

ASTACK    SEGMENT  STACK
            DW 12 DUP(?)    ; ОТВОДИТСЯ 12 СЛОВ ПАМЯТИ
ASTACK    ENDS

; ДАННЫЕ ПРОГРАММЫ

DATA      SEGMENT

;  ДИРЕКТИВЫ ОПИСАНИЯ ДАННЫХ

HELLO     DB 'ЗДРАВСТВУЙТЕ!', 0AH, 0DH,EOFLINE
GREETING  DB 'ВАС ПРИВЕТСТВУЕТ СТ.ГР. 1382 - РЫМАРЬ М.И.$'
```

```

DATA      ENDS

; КОД ПРОГРАММЫ

CODE      SEGMENT
          ASSUME CS:CODE, DS:DATA, SS:ASTACK

; 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 А

### ЛИСТИНГОВЫЕ ФАЙЛЫ

Название файла: *hello1.lst*

MICROSOFT (R) MACRO ASSEMBLER VERSION 5.10

9/13/22

00:17:37

PAGE

1-1

```

; HELLO1.ASM - УПРОЩЕННАЯ ВЕРСИЯ УЧЕБНОЙ
ПРОГР
АММЫ ЛАБ.РАБ. N1
;
ПО ДИСЦИПЛИНЕ "АРХИТЕКТУРА
КОМП
БЮТЕРА"
;
*****
*****
; НАЗНАЧЕНИЕ: ПРОГРАММА ФОРМИРУЕТ И ВЫВОДИТ
НА
ЭКРАН ПРИВЕТСТВИЕ
;
ПОЛЬЗОВАТЕЛЯ С ПОМОЩЬЮ ФУНКЦИИ
ДО
С "ВЫВОД СТРОКИ"
;
(НОМЕР 09 ПРЕРЫВАНИЕ 21Н),
КОТОРА
Я:
;
- ОБЕСПЕЧИВАЕТ ВЫВОД НА ЭКРАН
СТ
РОКИ СИМВОЛОВ,
;
ЗАКАНЧИВАЮЩЕЙСЯ ЗНАКОМ "$";
```

```

; - ТРЕБУЕТ ЗАДАНИЯ В РЕГИСТРЕ
АН
НОМЕРА ФУНКЦИИ=09H,
; А В РЕГИСТРЕ DX - СМЕЩЕНИЯ
А
ДРЕСА 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
0000 GREETING LABEL BYTE
; ТЕКСТ ПРИВЕТСТВИЯ
0000 82 A0 E1 20 AF E0 DB 'ВАС ПРИВЕТСТВУЕТ СТ.ГР.1381 -
РЫМАРЬ МАР

```

ИЯ.',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
90 EB AC A0 E0 EC
20 8C A0 E0 A8 EF
2E 0D 0A 24

```

```

.CODE ;

```

НАЧАЛ

О СЕГМЕНТА КОДА

```

0000 B8 ---- R          MOV  AX,@DATA          ;
ЗАГРУ
                                ЗКА В DS АДРЕСА НАЧАЛА
0003 8E D8              MOV  DS,AX          ;
СЕМЕ
                                НТА ДАННЫХ
0005 BA 0000 R          MOV  DX,OFFSET GREETING ;
ЗАГРУ
                                ЗКА В DX СМЕЩЕНИЯ
MICROSOFT (R) MACRO ASSEMBLER VERSION 5.10      9/13/22
00:17:37
                                PAGE
1-2
                                ;
АДРЕС
                                А ТЕКСТА ПРИВЕТСТВИЯ
0008                      DISPLAYGREETING:
0008 B4 09              MOV  AH,9          ;
# ФУН
                                КЦИИ ДОС ПЕЧАТИ СТРОКИ
000A CD 21              INT  21H          ;
ВЫВОД
                                НА ЭКРАН ПРИВЕТСТВИЯ
000C B4 4C              MOV  AH,4CH          ;
# ФУН
                                КЦИИ ДОС ЗАВЕРШЕНИЯ ПРОГРАММЫ
000E CD 21              INT  21H          ;
ЗАВЕР
                                ШЕНИЕ ПРОГРАММЫ И ВЫХОД В ДОС
                                END
MICROSOFT (R) MACRO ASSEMBLER VERSION 5.10      9/13/22
00:17:37
SYMBOLS-1
SEGMENTS AND GROUPS:

```

CLASS	N A M E	LENGTH	ALIGN	COMBINE
DGROUP	. . . . .	GROUP		
	_DATA . . . . .	002E 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

47992 + 461315 BYTES SYMBOL SPACE FREE

0 WARNING ERRORS

0 SEVERE ERRORS

Название файла: *hello2.lst*

Microsoft (R) Macro Assembler Version 5.10

9/13/22 01:09:23

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

0000          AStack      SEGMENT STACK
0000 000C[          DW 12 DUP(?)          ; Отводится 12 слов
п
              амяти
              ????
              ]

0018          AStack      ENDS

; Данные программы

0000          DATA      SEGMENT

; Директивы описания данных

0000 87 A4 E0 A0 A2 E1  HELLO      DB 'Здравствуйте!', 0AH, 0DH, EOFLine
      E2 A2 E3 A9 E2 A5
      21 0A 0D 24
0010 82 A0 E1 20 AF E0  GREETING DB 'Вас приветствует ст.гр. 1382 - Ры
      марь М.И.$'
      A8 A2 A5 E2 E1 E2
      A2 E3 A5 E2 20 E1
      E2 2E A3 E0 2E 20
      31 33 38 32 20 2D
      20 90 EB AC A0 E0
      EC 20 8C 2E 88 2E
      24
003B          DATA      ENDS

; Код программы

0000          CODE      SEGMENT
              ASSUME CS:CODE, DS:DATA, SS:AStack

; 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

Microsoft (R) Macro Assembler Version 5.10
9/13/22 01:09:23
Page 1-2

; Головная процедура
0005          Main      PROC FAR
0005 1E          push    DS          ;\ Сохранение адреса

```

```

0006  2B C0                      sub    AX,AX      ; > для последующего
В
                                остановки 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
                                рных.
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
Microsoft (R) Macro Assembler Version 5.10
9/13/22 01:09:23
Symbols-1

```

<i>N a m e</i>	<i>Length</i>	<i>Align</i>	<i>Combine Class</i>
<i>ASTACK</i> . . . . .	<i>0018</i>	<i>PARA</i>	<i>STACK</i>
<i>CODE</i> . . . . .	<i>001B</i>	<i>PARA</i>	<i>NONE</i>
<i>DATA</i> . . . . .	<i>003B</i>	<i>PARA</i>	<i>NONE</i>

<i>N a m e</i>	<i>Type</i>	<i>Value</i>	<i>Attr</i>
<i>EOFLINE</i> . . . . .		<i>NUMBER</i>	<i>0024</i>
<i>GREETING</i> . . . . .		<i>L BYTE</i>	<i>0010 DATA</i>
<i>HELLO</i> . . . . .		<i>L BYTE</i>	<i>0000 DATA</i>
<i>MAIN</i> . . . . .		<i>F PROC</i>	<i>0005 CODE Length = 0016</i>
<i>WRITEMSG</i> . . . . .		<i>N PROC</i>	<i>0000 CODE Length = 0005</i>
<i>@CPU</i> . . . . .		<i>TEXT</i>	<i>0101h</i>
<i>@FILENAME</i> . . . . .		<i>TEXT</i>	<i>hello2</i>
<i>@VERSION</i> . . . . .		<i>TEXT</i>	<i>510</i>

52 Total Lines

13 Symbols

48000 + 461307 Bytes symbol space free

0 Warning Errors

0 Severe Errors