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 Ассемблера.

Студент гр.0382 _____ Диденко Д.В.

Преподаватель _____ Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Научиться выполнять трансляцию, отладку и запускать программы на языке ассемблер.

Задание.

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.

Основные теоретические положения.

Лабораторная работа 1 использует 2 готовых программы на ассемблере: hello1 – составлена с использованием сокращенного описания сегментов и hello2 – составлена с полным описанием сегментов и выводом строки, оформленным как процедура. Выполнение работы состоит из двух частей.

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.

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

> afdpro hello1.exe

Выполнение работы.

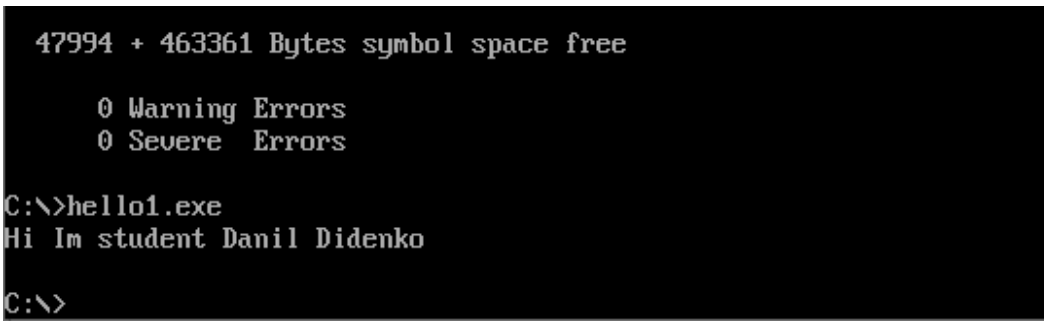
Результаты хода программы hello1.exe представлены в табл. 1.

Таблица 1 – Ход программы hello1.exe.

№ п/п	Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
				До выполнения	После выполнения
1.	0010	mov ax,1A07	B8071A	(AX) = 0000 (CS) = 1A05 (DS) = 19F5 (ES) = 19F5 (SS) = 1A09	(AX) = 1A07 (CS) = 1A05 (DS) = 19F5 (ES) = 19F5 (SS) = 1A09
2.	0013	Mov ds, ax	8ED8	(AX) = 1A07 (CS) = 1A05 (DS) = 19F5 (ES) = 19F5 (SS) = 1A09	(AX) = 1A07 (CS) = 1A05 (DS) = 1A07 (ES) = 19F5 (SS) = 1A09
3.	0015	Mov dx,0000	BA0000	(AX) = 1A07	(AX) = 1A07

				(CS) = 1A05 (DS) = 1A07 (ES) = 19F5 (SS) = 1A09	(CS) = 1A05 (DS) = 1A07 (ES) = 19F5 (SS) = 1A09
4.	0018	Mov ah,09	B409	(AX) = 1A07 (CS) = 1A05 (DS) = 1A07 (ES) = 19F5 (SS) = 1A09	(AX) = 0907 (CS) = 1A05 (DS) = 1A07 (ES) = 19F5 (SS) = 1A09
5.	001A	Int 21h	CD21	(AX) = 0907 (CS) = 1A05 (DS) = 1A07 (ES) = 19F5 (SS) = 1A09	(AX) = 0907 (CS) = 1A05 (DS) = 1A07 (ES) = 19F5 (SS) = 1A09
6.	001C	Mov ah,4ch	B44C	(AX) = 0907 (CS) = 1A05 (DS) = 1A07 (ES) = 19F5 (SS) = 1A09	(AX) = 4C07 (CS) = 1A05 (DS) = 1A07 (ES) = 19F5 (SS) = 1A09
7.	001E	Int 21	CD21	(AX) = 4C07 (CS) = 1A05 (DS) = 1A07 (ES) = 19F5 (SS) = 1A09	End

Результат работы программы hello1.exe.



Результаты хода программы hello2.exe представлены в табл. 2.

Таблица 2 – Ход программы hello2.exe.

№ п/п	Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
				До выполнения	После выполнения
1.	0005	Push DS	1E	(AX) = 0000 (CS) = 1A0B (DS) = 19F5 (ES) = 19F5 (SS) = 1A05 (SP) = 0000 Stack: +0: 0000 +2: 0000 +4: 0000 +6: 0000	(AX) = 0000 (CS) = 1A0B (DS) = 19F5 (ES) = 19F5 (SS) = 1A05 (SP) = 0016 Stack: +0: 19F5 +2: 0000 +4: 0000 +6: 0000
2.	0006	Sub ax,ax	2BCO	(AX) = 0000 (CS) = 1A0B (DS) = 19F5 (ES) = 19F5 (SS) = 1A05 (SP) = 0016 Stack: +0: 19F5 +2: 0000 +4: 0000 +6: 0000	(AX) = 0000 (CS) = 1A0B (DS) = 19F5 (ES) = 19F5 (SS) = 1A05 (SP) = 0016 Stack: +0: 19F5 +2: 0000 +4: 0000 +6: 0000
3.	0008	Push ax	50	(AX) = 0000 (CS) = 1A0B (DS) = 19F5 (ES) = 19F5 (SS) = 1A05 (SP) = 0016 Stack:	(AX) = 0000 (CS) = 1A0B (DS) = 19F5 (ES) = 19F5 (SS) = 1A05 (SP) = 0014 Stack:

				+0: 19F5 +2: 0000 +4: 0000 +6: 0000	+0: 0000 +2: 19F5 +4: 0000 +6: 0000
4.	0009	Mov ax,1A07(DATA)	B8071A	(AX) = 0000 (CS) = 1A0B (DS) = 19F5 (ES) = 19F5 (SS) = 1A05 (SP) = 0014 Stack: +0: 0000 +2: 19F5 +4: 0000 +6: 0000	(AX) = 1A07 (CS) = 1A0B (DS) = 19F5 (ES) = 19F5 (SS) = 1A05 (SP) = 0014 Stack: +0: 0000 +2: 19F5 +4: 0000 +6: 0000
5.	000C	Mov ds,ax	8ED8	(AX) = 1A07 (CS) = 1A0B (DS) = 19F5 (ES) = 19F5 (SS) = 1A05 (SP) = 0014 Stack: +0: 0000 +2: 19F5 +4: 0000 +6: 0000	(AX) = 1A07 (CS) = 1A0B (DS) = 1A07 (ES) = 19F5 (SS) = 1A05 (SP) = 0014 Stack: +0: 0000 +2: 19F5 +4: 0000 +6: 0000
6.	000E	Mov dx,0000 (OFFSET HELLO)	BA0000	(AX) = 1A07 (CS) = 1A0B (DS) = 19F5 (ES) = 19F5 (SS) = 1A05 (SP) = 0014 Stack: +0: 0000	(AX) = 1A07 (CS) = 1A0B (DS) = 19F5 (ES) = 19F5 (SS) = 1A05 (SP) = 0014 Stack: +0: 0000

				+2: 19F5 +4: 0000 +6: 0000	+2: 19F5 +4: 0000 +6: 0000
7.	0011	Call 0000 (WriteMsg)	E8ECFF	(AX) = 1A07 (CS) = 1A0B (DS) = 19F5 (ES) = 19F5 (SS) = 1A05 (SP) = 0014 Stack: +0: 0000 +2: 19F5 +4: 0000 +6: 0000	(AX) = 1A07 (CS) = 1A0B (DS) = 1A07 (ES) = 19F5 (SS) = 1A05 (SP) = 0012 Stack: +0: 0014 +2: 0000 +4: 19F5 +6: 0000
8.	0000	Mov ah,09	B409	(AX) = 1A07 (CS) = 1A0B (DS) = 1A07 (ES) = 19F5 (SS) = 1A05 (SP) = 0012 Stack: +0: 0014 +2: 0000 +4: 19F5 +6: 0000	(AX) = 0907 (CS) = 1A0B (DS) = 1A07 (ES) = 19F5 (SS) = 1A05 (SP) = 0012 Stack: +0: 0014 +2: 0000 +4: 19F5 +6: 0000
9.	0002	Int 21	CD21	(AX) = 0907 (CS) = 1A0B (DS) = 1A07 (ES) = 19F5 (SS) = 1A05 (SP) = 0012 Stack: +0: 0014 +2: 0000	(AX) = 0907 (CS) = 1A0B (DS) = 1A07 (ES) = 19F5 (SS) = 1A05 (SP) = 0012 Stack: +0: 0014 +2: 0000

				+4: 19F5 +6: 0000	+4: 19F5 +6: 0000
10.	0004	ret	C3	(AX) = 0907 (CS) = 1A0B (DS) = 1A07 (ES) = 19F5 (SS) = 1A05 (SP) = 0012 Stack: +0: 0014 +2: 0000 +4: 19F5 +6: 0000	(AX) = 0907 (CS) = 1A0B (DS) = 1A07 (ES) = 19F5 (SS) = 1A05 (SP) = 0014 Stack: +0: 0000 +2: 19F5 +4: 0000 +6: 0000
11.	0014	Mov dx, 0010	BA1000	(AX) = 0907 (CS) = 1A0B (DS) = 1A07 (ES) = 19F5 (SS) = 1A05 (SP) = 0014 Stack: +0: 0000 +2: 19F5 +4: 0000 +6: 0000	(AX) = 0907 (CS) = 1A0B (DS) = 1A07 (ES) = 19F5 (SS) = 1A05 (SP) = 0014 Stack: +0: 0000 +2: 19F5 +4: 0000 +6: 0000
12.	0017	Call 0000	E8E6FF	(AX) = 0907 (CS) = 1A0B (DS) = 1A07 (ES) = 19F5 (SS) = 1A05 (SP) = 0014 Stack: +0: 0000 +2: 19F5 +4: 0000	(AX) = 0907 (CS) = 1A0B (DS) = 1A07 (ES) = 19F5 (SS) = 1A05 (SP) = 0012 Stack: +0: 001A +2: 0000 +4: 19F5

				+6: 0000	+6: 0000
13.	0000	Mov ah,09	B409	(AX) = 0907 (CS) = 1A0B (DS) = 1A07 (ES) = 19F5 (SS) = 1A05 (SP) = 0012 Stack: +0: 001A +2: 0000 +4: 19F5 +6: 0000	(AX) = 0907 (CS) = 1A0B (DS) = 1A07 (ES) = 19F5 (SS) = 1A05 (SP) = 0012 Stack: +0: 001A +2: 0000 +4: 19F5 +6: 0000
14.	0002	Int 21	CD21	(AX) = 0907 (CS) = 1A0B (DS) = 1A07 (ES) = 19F5 (SS) = 1A05 (SP) = 0012 Stack: +0: 001A +2: 0000 +4: 19F5 +6: 0000	(AX) = 0907 (CS) = 1A0B (DS) = 1A07 (ES) = 19F5 (SS) = 1A05 (SP) = 0012 Stack: +0: 001A +2: 0000 +4: 19F5 +6: 0000
15.	0004	ret	C3	(AX) = 0907 (CS) = 1A0B (DS) = 1A07 (ES) = 19F5 (SS) = 1A05 (SP) = 0012 Stack: +0: 001A +2: 0000 +4: 19F5 +6: 0000	(AX) = 0907 (CS) = 1A0B (DS) = 1A07 (ES) = 19F5 (SS) = 1A05 (SP) = 0014 Stack: +0: 0000 +2: 19F5 +4: 0000 +6: 0000

16.	001A	Ret far	CB	(AX) = 0907 (CS) = 1A0B (DS) = 1A07 (ES) = 19F5 (SS) = 1A05 (SP) = 0014 Stack: +0: 0000 +2: 19F5 +4: 0000 +6: 0000	(AX) = 0907 (CS) = 19F5 (DS) = 1A07 (ES) = 19F5 (SS) = 1A05 (SP) = 0018 Stack: +0: 0000 +2: 0000 +4: 0000 +6: 0000
17.	0000	Int 20	CD20	(AX) = 0907 (CS) = 19F5 (DS) = 1A07 (ES) = 19F5 (SS) = 1A05 (SP) = 0018 Stack: +0: 0000 +2: 0000 +4: 0000 +6: 0000	end

Результат работы программы hello2.exe

```
C:\>hello2.exe
Hello Worlds!
Student from 0382 - Danil Didenko
C:\>
```

Выводы.

Выполнены трансляция, отладка и запуск программы на языке ассемблер.

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 'Hi Im student Danil Didenko',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                  9/15/21
08:46:47
                                                          Page     1-1


                          DOSSEG
                          .MODEL  SMALL
                          .STACK  100h
                          .DATA
      0000                  Greeting  LABEL  BYTE
      0000  48 69 20 49 6D 20       DB    'Hi    Im    student    Danil
Didenko',13,10,'$'
            73 74 75 64 65 6E
            74 20 44 61 6E 69
            6C 20 44 69 64 65
```

```
       6E 6B 6F 0D 0A 24
                    .CODE
 0000  B8 ---- R         mov  ax, @data
 0003  8E D8                mov  ds, ax
 0005  BA 0000 R         mov  dx, OFFSET Greeting

 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 9/15/21 08:46:47

Symbols-1

Segments and Groups:

N a m e	Length	Align	Combine	Class
DGROUP	GROUP			
_DATA	001E	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


     18 Source  Lines
     18 Total   Lines
     19 Symbols

  47994 + 463361 Bytes symbol space free

      0 Warning Errors
 0 Severe  Errors
```

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 0382 - Danil Didenko$'
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                  9/15/21
11:28:21
                                                            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 0382 -
Danil Didenko$'
      74 20 66 72 6F 6D
      20 30 33 38 32 20
      2D 20 44 61 6E 69
      6C 20 44 69 64 65
      6E 6B 6F 24
0032                DATA      ENDS


0000                CODE      SEGMENT

0000                WriteMsg  PROC  NEAR
0000  B4 09                        mov   AH,9
0002  CD 21                        int   21h
0004  C3                    ret
0005                WriteMsg  ENDP
```

```
                        ; Головная процедура
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
```

Microsoft (R) Macro Assembler Version 5.10
9/15/21 11:28:21

Symbols-1

Segments and Groups:

N a m e	Length	Align	Combine	Class
ASTACK	0018	PARA	STACK	
CODE	001B	PARA	NONE	
DATA	0032	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


     47 Source  Lines
     47 Total   Lines
     13 Symbols

  47986 + 461321 Bytes symbol space free

      0 Warning Errors
 0 Severe  Errors
```