

**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. 0384

\_\_\_\_\_

Кусмарцев А.И.

Преподаватель

\_\_\_\_\_

Ефремов М.А.

Санкт-Петербург

2021

### Цель работы.

Рассмотреть работу программ на языке Ассемблера. Понять структуру и реализацию программ.

### Выполнение работы.

1) Транслируем *hello1.asm* и *hello2.asm* с помощью MASM, для получения объектных модулей *hello1.asm* и *hello2.asm*. В диагностических файлах ошибок не обнаружено.

2) Компоуем загрузочные модули с помощью LINK.

3) Выполняем программы в автоматическом режиме. Вывод *hello1.exe*:

```
D:\>hello1.exe
Greetings from student from the group 0383 - Kusmartsev A.I.
```

Вывод *hello2.exe*:

```
D:\>hello2.exe
Hello Worlds!
Student from 0383 - Kusmartsev A.I.
```

4) Запустим выполнение программ под управлением отладчика AFDPRO.

5) Начальное содержание сегментных регистров программы *hello1.exe*.

(CS)=1A05

(DS)=19F5

(ES)=19F5

(SS)=1A0C

Таблица 1 – Таблица результатов прогона программы *hello1.exe* под управлением отладчика

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) = 0000 (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

15	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 Программа завершилась

6) Начальное содержание сегментных регистров программы hello2.exe.

(CS)=1A0B

(DS)=19F5

(ES)=19F5

(SS)=1A05

Таблица 2 – Таблица результатов прогона программы hello2.exe под управлением отладчика

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 (CS) = 1A0A (SP) = 0018 Stack +0 0000 (IP) = 0010	(AX) = 1A07 (DX) = 0000 (DS) = 19F5 (CS) = 1A0A (SP) = 0016 Stack +0 19F5 (IP) = 0013
0006	Sub AX,AX	2BC0	(AX) = 0000 (DX) = 0000 (DS) = 19F5 (CS) = 1A0A (SP) = 0016 Stack +0 19F5 (IP) = 0006	(AX) = 0000 (DX) = 0000 (DS) = 19F5 (CS) = 1A0A (SP) = 0016 Stack +0 19F5 (IP) = 0008

0008	Push AX	50	(AX) = 0000 (DX) = 0000 (DS) = 19F5 (CS) = 1A0A (SP) = 0016 Stack +0 19F5 (IP) = 0008	(AX) = 0000 (DX) = 0000 (DS) = 19F5 (CS) = 1A0A (SP) = 0014 Stack +0 0000 Stack +2 19F5 (IP) = 0009
0009	Mov AX,1A07	B8071A	(AX) = 0000 (DX) = 0000 (DS) = 19F5 (CS) = 1A0A (SP) = 0014 Stack +0 0000 Stack +2 19F5 (IP) = 0009	(AX) = 1A07 (DX) = 0000 (DS) = 19F5 (CS) = 1A0A (SP) = 0014 Stack +0 0000 Stack +2 19F5 (IP) = 000C
000C	Mov DS,AX	8ED8	(AX) = 1A07 (DX) = 0000 (DS) = 19F5 (CS) = 1A0A (SP) = 0014 Stack +0 0000 Stack +2 19F5 (IP) = 000C	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (CS) = 1A0A (SP) = 0014 Stack +0 0000 Stack +2 19F5 (IP) = 000E
000E	Mov DX,0000	BA0000	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (CS) = 1A0A (SP) = 0014 Stack +0 0000 Stack +2 19F5 (IP) = 000E	(AX) = 1A07 (DX) = 0000 (DS) = 1A07( (CS) = 1A0A (SP) = 0014 Stack +0 0000 Stack +2 19F5 (IP) = 0011
0011	CALL 0000	E8ECFF	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (CS) = 1A0A (SP) = 0014 Stack +0 0000 Stack +2 19F5 (IP) = 0011	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (CS) = 1A0A (SP) = 0012 Stack +0 0014 Stack +2 0000 Stack +4 19F5 (IP) = 0000
0000	Mov AH,09	B409	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (CS) = 1A0A (SP) = 0012 Stack +0 0014	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (CS) = 1A0A (SP) = 0012 Stack +0 0014

			Stack +2 0000 Stack +4 19F5 (IP) = 0000	Stack +2 0000 Stack +4 19F5 (IP) = 0002
0002	Int 21	CD21	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (CS) = 1A0A (SP) = 0012 Stack +0 0014 Stack +2 0000 Stack +4 19F5 (IP) = 0002	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (CS) = 1A0A (SP) = 0012 Stack +0 0014 Stack +2 0000 Stack +4 19F5 (IP) = 0004
0004	Ret	C3	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (CS) = 1A0A (SP) = 0012 Stack +0 0014 Stack +2 0000 Stack +4 19F5 (IP) = 0004	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (CS) = 1A0A (SP) = 0014 Stack +0 0000 Stack +2 19F5 (IP) = 00014
0014	Mov DX,0010	BA1000	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (CS) = 1A0A (SP) = 0014 Stack +0 0000 Stack +2 19F5 (IP) = 0014	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (CS) = 1A0A (SP) = 0014 Stack +0 0000 Stack +2 19F5 (IP) = 0017
0017	Call 0000	E8E6FF	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (CS) = 1A0A (SP) = 0014 Stack +0 0000 Stack +2 19F5 (IP) = 0017	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (CS) = 1A0A (SP) = 0012 Stack +0 001A Stack +2 0000 Stack +4 19F5 (IP) = 0000
0000	Mov AH,09	B409	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (CS) = 1A0A (SP) = 0012 Stack +0 001A Stack +2 0000 Stack +4 19F5 (IP) = 0000	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (CS) = 1A0A (SP) = 0012 Stack +0 001A Stack +2 0000 Stack +4 19F5 (IP) = 0002

0002	Int 21	CD21	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (CS) = 1A0A (SP) = 0012 Stack +0 001A Stack +2 0000 Stack +4 19F5 (IP) = 0002	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (CS) = 1A0A (SP) = 0012 Stack +0 001A Stack +2 0000 Stack +4 19F5 (IP) = 0004
0004	Ret	C3	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (CS) = 1A0A (SP) = 0012 Stack +0 001A Stack +2 0000 Stack +4 19F5 (IP) = 0004	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (CS) = 1A0A (SP) = 0014 Stack +0 0000 Stack +2 0000 (IP) = 001A
001A	Ret Far	CB	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (CS) = 1A0A (SP) = 0014 Stack +0 0000 Stack +2 0000 (IP) = 001A	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (CS) = 19F5 (SP) = 0018 Stack +0 0000 (IP) = 0000
0000	Int 20	CD20	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (CS) = 19F5 (SP) = 0018 Stack +0 0000 (IP) = 0000	(AX) = 0000 (DS) = 19F5 (IP) = 0005 Программа завершилась

### **Выводы.**

В ходе выполнения лабораторной работы была рассмотрена работа с эмулятором Dosbox, были рассмотрены простейшие программы на языке Ассемблера.

# ПРИЛОЖЕНИЕ А

## ТЕКСТЫ ИСХОДНЫХ ФАЙЛОВ ПРОГРАММ

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 'Greetings from student from the group 0383 - Kusmartsev
A.I.',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
        ; 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 0383 - Кусмарцев А.И.$'
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 Б

## ТЕКСТЫ ФАЙЛОВ ДИАГНОСТИЧЕСКИХ СООБЩЕНИЙ

### ПРОГРАММ

Название файла: hello1.lst  
 Microsoft (R) Macro Assembler Version 5.10  
 9/13/21 15:02:25  
 1-1  
 Page

```

; HELLO1.ASM - упрощенная версия учебной программы лаб.р
; 06. N1
; по дисциплине "Архитектура компьютера"
;
*****
*****
; Назначение: Программа формирует и выводит на экран
; приветствие
; пользователя с помощью функции DOS "Вывод строки"
; (номер 09 прерывания 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 DOS
C
.MODEL SMALL
; Модель памяти-SMALL(Малая)
.STACK 100h
; Отвести под Стек 256 байт
  
```

```

                .DATA
                ; Начало сегмента данных
0000          Greeting LABEL BYTE
                ; Текст приветствия
0000 47 72 65 65 74 69      DB 'Greetings from student from the
group 0383 - Kusmartsev A.I.'$'
                6E 67 73 20 66 72
                6F 6D 20 73 74 75
                64 65 6E 74 20 66
                72 6F 6D 20 74 68
                65 20 67 72 6F 75
                70 20 30 33 38 33
                20 2D 20 4B 75 73

```

```

Microsoft      (R)      Macro      Assembler      Version      5.10
9/13/21 15:02:25

```

Page

1-2

```

6D 61 72 74 73 65
76 20 41 2E 49 2E
0D 0A 24

```

```

                .CODE
;
Над
0000  B8 ---- R      ; начало сегмента кода
3a      mov ax, @data
;
0003  8E D8          ; грузка в DS адреса начала
; ce      mov ds, ax
;
0005  BA 0000 R      ; гмента данных
3a      mov dx, OFFSET Greeting
;
; ад      ; грузка в dx смещения
;
0008          ; аса текста приветствия
0008  B4 09          DisplayGreeting:
; # ф      mov ah, 9
;
000A  CD 21          ; функции ДОС печати строки
; вы      int 21h
;
000C  B4 4C          ; од на экран приветствия
; # ф      mov ah, 4ch
;
000E  CD 21          ; функции ДОС завершения про
; за      ; граммы
;      int 21h
;
;      ; ершение программы и выход
;      ; в ДОС
;      END

```

```

Microsoft      (R)      Macro      Assembler      Version      5.10
9/13/21 15:02:25

```

ols-1

## Segments and Groups:

Class	N a m e	Length	Align	Combine
DGROUP	. . . . .	GROUP		
_DATA	. . . . .	004C	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 + 459266 Bytes symbol space free

0 Warning Errors  
 0 Severe Errors

```

; HELLO2 - Учебная программа N2
; лаб.раб.#1 по дисциплине "Архитектура компьютера"
; Программа использует процедуру для печати строки
;
;      ТЕКСТ ПРОГРАММЫ

= 0024      EOFLine EQU '$' ; Определена символическая константа "Конец строки"

; Стек программы
ASSUME CS:CODE, SS:ASStack

0000      ASStack      SEGMENT      STACK
0000      0000C[      DW 12 DUP('!') ;
ОТВОДИТСЯ      12 слов памяти
0021      ]

0018      ASStack      ENDS

; Данные программы

0000      DATA      SEGMENT

; Директивы описания данных
0000      48 65 6C 6C 6F 20      HELLO      DB 'Hello Worlds!', 0AH,
0DH, EOFLine
0000      57 6F 72 6C 64 73
0000      21 0A 0D 24
0010      53 74 75 64 65 6E      GREETING      DB 'Student from 0383 -
Kusmartsev A.I.
$'
0000      74 20 66 72 6F 6D
0000      20 30 33 38 33 20
0000      2D 20 4B 75 73 6D
0000      61 72 74 73 65 76
0000      20 41 2E 49 2E 24
0033      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 прерыванию
0004 C3                                ret
Microsoft (R) Macro Assembler Version 5.10
9/13/21 14:36:38
1-2
Page

```

```

0005                                WriteMsg ENDP

                                ; Головная процедура
0005                                Main PROC FAR
0005 1E                                push     DS ;\ Сохранение
                                е адреса начала PSP в стеке
0006 2B C0                                sub      AX,AX ; > для
после
                                ующего восстановления по
0008 50                                push     AX ;/ команде ret
                                , завершающей процедуру.
0009 B8 ---- R                                mov     AX,DATA ; Загрузка
                               зка сегментного
000C 8E D8                                mov     DS,AX ;
реги
                                тра данных.
000E BA 0000 R                                mov     DX, OFFSET HELLO ; Вывод
                                на экран первой
0011 E8 0000 R                                call    WriteMsg ; строка
                                и приветствия.
0014 BA 0010 R                                mov     DX, OFFSET GREETING ; Вывод
                                на экран второй
0017 E8 0000 R                                call    WriteMsg ; строка
                                и приветствия.
001A CB                                ret ; Выход
                                в DOS по команде,
                                ; находящейся в 1-ом слове PSP.
001B                                Main ENDP
001B                                CODE ENDS
                                END Main
Microsoft (R) Macro Assembler Version 5.10
9/13/21 14:36:38
Symb
ols-1

```

#### Segments and Groups:

Class	N a m e	Length	Align	Combine
-------	---------	--------	-------	---------

ASTACK . . . . .	0018	PARA	STACK
CODE . . . . .	001B	PARA	NONE
DATA . . . . .	0033	PARA	NONE

Symbols:

	N a m e	Type	Value	Attr
	EOFLINE . . . . .	NUMBER		0024
	GREETING . . . . .	L BYTE		0010 DATA
	HELLO . . . . .	L BYTE		0000 DATA
= 0016	MAIN . . . . .	F PROC		0005 CODE Length
= 0005	WRITEMSG . . . . .	N PROC		0000 CODE Length
	@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