

**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**

Студентка гр. 1381

\_\_\_\_\_

Тулегенова А.О.

Преподаватель

\_\_\_\_\_

Ефремов М.А.

Санкт-Петербург

2022

## Цель работы.

Изучить основные принципы трансляции, отладки и выполнения программ на языке Ассемблера.

## Задание

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 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*

8. Выполнить пункты 1 - 7 части 1 настоящего задания применительно к программе hello2.asm.

### **Ход выполнения работы**

В ходе выполнения лабораторной работы были загружены файлы hello1.asm и hello2.asm в каталог masm. Далее программы были протранслированы с созданием объектного файла и файла диагностических сообщений. Был скомпонован загрузочный файл с созданием исполняемого файла. Далее была выполнена программа в автоматическом режиме. Была произведена отладка программы в пошаговом режиме с помощью отладчика.

Таблица 1. Начальное значение регистров

CS	DS	ES	SS
1A05	19F5	19F5	1A0C

Таблица 2. Протокол работы программы hello1

Адрес команды	Символьный 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 (IP)=0015 (DX)=0000	(AX)=1A07 (DS)=1A07 (IP)=0018 (DX)=0000
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

Таблица 3. Начальное значение регистров

CS	DS	ES	SS
1A05	19F5	19F5	1A0C

Таблица 4. Протокол работы программы hello2

Адрес команды	Символьный код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0005	PUSH DS	1E	(AX)=0000 (DS)=19F5 (IP)=0005 Stack +0 0000 +2 0000 +4 0000 +6 0000	(AX)=0000 (DS)=19F5 (IP)=0006 Stack +0 19F5 +2 0000 +4 0000 +6 0000
0006	SUB AX, AX	2BC0	(AX)=0000 (DS)=19F5 (IP)=0006	(AX)=0000 (DS)=19F5 (IP)=0008
0008	PUSH AX	50	(AX)=0000 (DS)=19F5 (IP)=0008 Stack +0 19F5 +2 0000 +4 0000 +6 0000	(AX)=0000 (DS)=19F5 (IP)=0009 Stack +0 0000 +2 19F5 +4 0000 +6 0000
0009	MOV AX, 1A07	B8071A	(AX)=0000 (DS)=19F5 (IP)=0009	(AX)=1A07 (DS)=19F5 (IP)=000C
000C	MOV DS, AX	8ED8	(AX)=1A07 (DS)=19F5 (IP)=000C	(AX)=1A07 (DS)=1A07 (IP)=000E
000E	MOV DX, 0000	BA0000	(AX)=1A07 (DS)=1A07 (IP)=000E (DX)=0000	(AX)=1A07 (DS)=1A07 (IP)=0011 (DX)=0000

0011	CALL 0000	E8ECFF	(AX)=4C07 (DS)=1A07 (IP)=001E Stack +0 0000 +2 19F5 +4 0000 +6 0000	(AX)=1A07 (DS)=1A07 (IP)=0000 Stack +0 0014 +2 0000 +4 19F5 +6 0000
0000	MOV AH, 09	B409	(AX)=1A07 (DS)=1A07 (IP)=0000	(AX)=0907 (DS)=1A07 (IP)=0002
0002	INT 21	CD21	(AX)=0907 (DS)=1A07 (IP)=0002	(AX)=0907 (DS)=1A07 (IP)=0004
0004	RET	C3	(AX)=0907 (DS)=1A07 (IP)=0004 Stack +0 0014 +2 0000 +4 19F5 +6 0000	(AX)=0907 (DS)=1A07 (IP)=0014 Stack +0 0000 +2 19F5 +4 0000 +6 0000
0014	BA1000	MOV DX, 0010	(AX)=0907 (DS)=1A07 (IP)=0014 (DX)=0000	(AX)=0907 (DS)=1A07 (IP)=0017 (DX)=0010
0017	E8E6FF	CALL 0000	(AX)=0907 (DS)=1A07 (IP)=0017 Stack +0 0000 +2 19F5 +4 0000 +6 0000	(AX)=0907 (DS)=1A07 (IP)=0000 Stack +0 001A +2 0000 +4 19F5 +6 0000
0000	MOV AH, 09	B409	(AX)=0907 (DS)=1A07 (IP)=0000	(AX)=0907 (DS)=1A07 (IP)=0002
0002	INT 21	CD21	(AX)=0907 (DS)=1A07 (IP)=0002	(AX)=0907 (DS)=1A07 (IP)=0004
0004	RET	C3	(AX)=0907 (DS)=1A07 (IP)=0004 Stack +0 001A +2 0000 +4 19F5 +6 0000	(AX)=0907 (DS)=1A07 (IP)=001A Stack +0 0000 +2 19F5 +4 0000 +6 0000

001A	RET Far	CB	(AX)=0907 (DS)=1A07 (IP)=001A Stack +0 0000 +2 19F5 +4 0000 +6 0000	(AX)=0907 (DS)=1A07 (IP)=0000 Stack +0 0000 +2 0000 +4 0000 +6 0000
0000	INT 20	CD20	(AX)=0907 (DS)=1A07 (IP)=0000 (CS)=19F5	(AX)=00000 (DS)=19F5 (IP)=0005 (CS)=1A0A

### Вывод

При выполнении лабораторной работы были изучены основные этапы создания программы на языке Assembler: трансляция, отладка и выполнение программы.

## ПРИЛОЖЕНИЕ А

### ТЕКСТЫ ИСХОДНЫХ ФАЙЛОВ

Hello1.asm:

```
DOSSEG
.MODEL    SMALL
.STACK    100h
.DATA
Greeting LABEL BYTE                ; Текст
приветствия
    DB 'Вас приветствует ст.гр.1381 - Тулегенова
A.O.',13,10,'$'
.CODE
    mov ax, @data
    mov ds, ax
    mov dx, OFFSET Greeting

DisplayGreeting:
    mov ah, 9h
    int 21h
    mov ah, 4ch
    int 21h
END
```

Hello2.asm:

```
EOFLine EQU '$'
ASSUME CS:CODE, SS:AStack

AStack    SEGMENT    STACK
          DW 12 DUP('!')
AStack    ENDS

DATA      SEGMENT

HELLO     DB 'Hello Worlds!', 0AH, 0DH,EOFLine
GREETING  DB 'Student from 1381 - $'
DATA      ENDS

CODE      SEGMENT
WriteMsg  PROC    NEAR
          mov     AH,9
          int     21h
          ret
WriteMsg  ENDP

Main      PROC    FAR
          push    DS
```

```

sub    AX,AX
push   AX
mov    AX,DATA
mov    DS,AX
mov    DX, OFFSET HELLO
call   WriteMsg
mov    DX, OFFSET GREETING
call   WriteMsg
ret

Main   ENDP
CODE   ENDS
END Main

```





```

        .STACK 100h
        ; PhC,PIPµCfC,Pë PîPsPr PŸC,PµPe 256 P±P°PN°C,
        .DATA
        ; PkP°C±P°P»Ps CÍPµPiPjPµPSC,P° PrP°PSPSC<C...
0000      Greeting LABEL BYTE
        ; PŸPµPeCfC, PîCßPëPIPIµC,CfC,PIPëCŰ
0000  D0 92 D0 B0 D1 81      DB 'P'P°CÍ PîCßPëPIPIµC,CfC,PICfPµC,
CfC,.Pic
        B.7303 - P□PIP°PSPsPI P□.P□.',13,10,'$'
        20 D0 BF D1 80 D0
        B8 D0 B2 D0 B5 D1
        82 D1 81 D1 82 D0
        B2 D1 83 D0 B5 D1
        82 20 D1 81 D1 82
        2E D0 B3 D1 80 2E
        37 33 30 33 20 2D

```

Microsoft (R) Macro Assembler Version 5.10  
9/11/22 20:05:15

Page

1-2

```

        20 D0 98 D0 B2 D0
        B0 D0 BD D0 BE D0
        B2 20 D0 98 2E D0
        98 2E 0D 0A 24
        .CODE
        ; PkP°C
        ;P°P»Ps CÍPµPiPjPµPSC,P° PePsPrP°
0000  B8 ---- R      mov ax, @data
P-P°
        PîCßCfP·PeP° PI DS P°PrCßPµCfP° PSP°C±P°P»P°
0003  8E D8          mov ds, ax
CÍPµ
        PiPjPµPSC,P° PrP°PSPSC<C...
0005  BA 0000 R      mov dx, OFFSET Greeting
P-P°
        PîCßCfP·PeP° PI dx CÍPjPµC%PµPSPëCŰ
        ; P°PrC
        BµPµCfP° C,PµPeCfC,P° PîCßPëPIPIµC,CfC,PIPëCŰ
0008      DisplayGreeting:
0008  B4 09          mov ah, 9h
; # C
        „CfPSPeC†PëPë P”PhPŸ PîPµC±P°C,Pë CfC,CßPsPePë
000A  CD 21          int 21h
PIC<P
        IPsPr PSP° CÍPeCßP°PS PîCßPëPIPIµC,CfC,PIPëCŰ
000C  B4 4C          mov ah, 4ch
# C,,
        CfPSPeC†PëPë P”PhPŸ P·P°PIPIµCßCëPµPSPëCŰ PîCßPs
        PîCßP°PjPjC<
000E  CD 21          int 21h
P·P°P
        IPµCßCëPµPSPëPµ PîCßPsPîCßP°PjPjC< Pë PIC<C...PsP

```

r PI P"PhPř

END

Microsoft (R) Macro Assembler Version 5.10  
9/11/22 20:05:15

Symbols-1

Segments and Groups:

Class	N a m e	Length	Align	Combine
DGROUP	. . . . .	GROUP		
__DATA	. . . . .	0047 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.lst:

Microsoft (R) Macro Assembler Version 5.10  
9/11/22 20:07:10

Page 1-1

```

; HELLO2 - PJC+PμP±PSP°Cμ
PìCḂPsPìCḂP°PjPjP° N2
P»P°P±.CḂP°P±.#1 PìPs
PrPëCíC†PëPìP»PëPSPμ "P
ḡCḂC...PëC, PμPeC, CíCḂP°
PePsPjPìCḂCḂC, PμCḂP°"
; PμCḂPsPìCḂP°PjPjP°
PëCíPìPsP»CḂP·CíP
μC, PìCḂPsC†PμPrCíCḂCí PrP»Cμ PìPμC†P°C, Pë
CíC,
CḂPsPePë
;
; PŸP•PḂPŸPŸ PμP PḡP`P PḡPḂPḂP«
= 0024 EOFLine EQU '$' ;
PḡPìCḂPμPrPμP»PμPSP
ëPμ CíPëPjPìPsP»CḂPSPsPN°
PePsPSCíC, P°PSC, C<
;
"PḂPsPSPμC† CíC
, CḂPsPePë"
; PŸC, PμPe PìCḂPsPìCḂP°PjPjC<
ASSUME CS:CODE, SS:AStack
0000 AStack SEGMENT STACK
0000 000C[ DW 12 DUP('!') ;
PḡC, PìPsPrPëC, CíC
μ 12 CíP»PsPì PìP°PjCμC, Pë
0021
]
0018 AStack ENDS
; P`P°PSPSC<Pμ PìCḂPsPìCḂP°PjPjC<
0000 DATA SEGMENT

```

```

; P"PeCbPµPeC,PePIC< PsPiPeCfP°PSPeCµ
PrP°PSPS
C<C...

```

```

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 - $'
74 20 66 72 6F 6D
20 34 33 35 30 20
2D 20 24
0025 DATA ENDS

```

```

; PµPsPr PiCbPsPiCbP°PjPjC<

0000 CODE SEGMENT
; PµCbPsC†PµPrCfCbP° PiPµC†P°C,Pe
CfC,CbPsPePe
0000 WriteMsg PROC NEAR
0000 B4 09 mov AH,9
0002 CD 21 int 21h ; P'C<P·PsPI
C,,CfPSPeC†PeP
ë DOS PiPs PiCbPµCbC<PIP°PSPeCb
0004 C3 ret
0005 WriteMsg ENDP

```

```

; P`PsP»PsPIPSP°Cµ PiCbPsC†PµPrCfCbP°
Microsoft (R) Macro Assembler Version 5.10
9/11/22 20:07:10

```

Page 1-2

```

0005 Main PROC FAR
0005 1E push DS ;\
PŸPsC...CbP°PSPµPSPe
Pµ P°PrCbPµCfP° PSP°C†P°P»P° PSP PI
CfC,PµPePµ
0006 2B C0 sub AX,AX ; > PrP»Cµ
PiPsCfP»PµP

```

```

rCfCtC%PµPiPs PIPsCfCfC,P°PSPsPIP»PµPSPëCµ
PiPs
0008 50 push AX ;/
PePsPjP°PSPrPµ ret
, P·P°PIPµCtCëP°CtC%PµPN
PiCtPsC†PµPrCfCtCf.
0009 B8 ---- R mov AX, DATA ;
P-P°PiCtC
fP·PeP° CfPµPiPjPµPSC,PSPsPiPs
000C 8E D8 mov DS,AX ;
CtPµPiPëC
fC,CtP° PrP°PSPSC<C...
000E BA 0000 R mov DX, OFFSET HELLO ;
P'C<PIPsP
r PSP° CkPeCtP°PS PiPµCtPIPsPN
0011 E8 0000 R call WriteMsg ;
CfC,CtPsP
ePë PiCtPëPIPµC,CfC,PIPëCµ.
0014 BA 0010 R mov DX, OFFSET GREETING ;
P'C<PIPsP
r PSP° CkPeCtP°PS PIC,PsCtPsPN
0017 E8 0000 R call WriteMsg ;
CfC,CtPsP
ePë PiCtPëPIPµC,CfC,PIPëCµ.
001A CB ret ;
P'C<C...PsP
r PI DOS PiPs PePsPjP°PSPrPµ,
;
PSP°C...PsP
rCµC%PµPN:CfCµ PI 1-PsPj CfP»PsPIPµ PSP.
001B Main ENDP
001B CODE ENDS
END Main
Microsoft (R) Macro Assembler Version 5.10
9/11/22 20:07:10

```

Symbols-1

Segments and Groups:

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

ASTACK	. . . . .	0018	PARA	STACK
CODE	. . . . .	001B	PARA	NONE
DATA	. . . . .	0025	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