

**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. 1303

Самохин К.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

## **Цель работы.**

Изучить и применить на практике трансляцию, отладку и выполнение 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

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.

Результаты прогона программы под управлением отладчика должны быть представлены в виде, показанном на примере одной команды в табл.1, и подписаны преподавателем.

Адрес Команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			до выполнения .	После выполнения
0003	Mov DS, AX	8E D8	(AX) = 2D87 (DS) = 2D75 (IP) = 0003	(AX) = 2D87 (DS) = 2D87 (IP) = 0005

### **Выполнение работы.**

Часть 1:

1. В текстовом редакторе была просмотрен код программы hello1.asm
2. Строка — приветствие была преобразована в соответствии с моими личными данными, каждый сегмент программы был разобран.
3. Файл hello1.asm был загружен в папку DOSBOX в эмуляторе MS-DOS

4. Программа была протранслирована с помощью команды:

> masm hello1.asm

5. Скомпонован загрузочный модуль с созданием карты памяти и исполняемого файла hello1.exe. с помощью строки:

> link hello1.obj

6. Программа выполнена в автоматическом режиме путем набора строки:

> hello1.exe

Программа выполнена корректно

7. Программа была запущена под управлением отладчика AFDPRO:

> afdpro hello1.exe

Начальные значения регистров:

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	B8AE11	(AX) = 0000 (IP) = 0010	(AX) = 1A07 (IP) = 0013

0013	MOV DS, AX	8ED8	(DS) = 19F5 (IP) = 0013 (AX) = 1A07	(DS) = 1A07 (IP) = 0015 (AX) = 1A07
0015	MOV DX, 0000	BA0000	(IP) = 0015	(IP) = 0018
0018	MOV AH, 9	B409	(AX) = 1A07 (AH) = 1A (IP) = 0018	(AX) = 0907 (AH) = 09 (IP) = 001A
001A	INT 21H	CD21	(IP) = 001A	(IP) = 001C
001C	MOV AH, 4CH	B44C	(AH) = 09 (AX) = 0907 (IP) = 001C	(AH) = 4C (AX) = 4C07 (IP) = 001E
001E	INT 21H	CD21	(AX) = 4C07 (DS) = 1A07 (IP) = 001E	(AX) = 0000 (DS) = 19F5 (IP) = 0010

## Часть 2.

1. В текстовом редакторе была просмотрен код программы hello2.asm2
2. Строка — приветствие была преобразована в соответствии с моими личными данными, каждый сегмент программы был разобран.
3. Файл hello2.asm был загружен в папку DOSBOX в эмуляторе MS-DOS
4. Программа была протранслирована с помощью команды:

```
> masm hello2.asm
```

5. Скомпонован загрузочный модуль с созданием карты памяти и исполняемого файла hello2.exe. с помощью строки:

```
> link hello2.obj
```

6. Программа выполнена в автоматическом режиме путем набора строки:

```
> hello2.exe
```

Программа выполнена корректно

7. Программа была запущена под управлением отладчика AFDPRO:

```
> afdpro hello2.exe
```

Начальные значения регистров:

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	(SP) = 0018 (IP) = 0005 Stack +0 = 0000	(SP) = 0016 (IP) = 0006 Stack +0 = 19F5
0006	SUB AX,AX	2BC0	(IP) = 0006 (AX) = 0000 (ZF) = 0 (PF) = 0	(IP) = 0008 (AX) = 0000 (ZF) = 1 (PF) = 1
0008	PUSH AX	50	(SP) = 0016 (IP) = 0008 Stack +0 = 19F5 Stack +2 = 0000	(SP) = 0014 (IP) = 0009 Stack +0 = 0000 Stack +2 = 19F5
0009	MOV AX,DATA	B8AE11	(AX) = 0000 (IP) = 0009	(AX) = 1A07 (IP) = 000C
000C	MOV DS,AX	8ED8	(DS) = 19F5 (IP) = 000C	(DS) = 1A07 (IP) = 000E

000E	MOV DX, 0000	BA0000	(DX) = 0000 (IP) = 000E	(DX) = 0000 (IP) = 0011
0011	CALL 0000	E8ECFF	(SP) = 0014 (IP) = 0011 Stack +0 = 0000 Stack +2 = 19F5 Stack +4 = 0000	(SP) = 0012 (IP) = 0000 Stack +0 = 0014 Stack +2 = 0000 Stack +4 = 19F5
0000	MOV AH,9	B409	(AH) = 1A (AX) = 1A07 (IP) = 0000	(AH) = 09 (AX) = 0907 (IP) = 0002
0002	INT 21H	CD21	(IP) = 0002	(IP) = 0004
0004	RET	C3	(SP) = 0012 (IP) = 0004 Stack +0 = 0014 Stack +2 = 0000 Stack +4 = 19F5	(SP) = 0014 (IP) = 0014 Stack +0 = 0000 Stack +2 = 19F5 Stack +4 = 0000



0014	MOV DX, 0010	BA0D00	(DX) = 0000 (IP) = 0014	(DX) = 0010 (IP) = 0017
0017	CALL 0000	E8E6FF	(SP) = 0014 (IP) = 0017 Stack +0 = 0000 Stack +2 = 19F5 Stack +4 = 0000	(SP) = 0012 (IP) = 0000 Stack +0 = 001A Stack +2 = 0000 Stack +4 = 19F5
0000	MOV AH,9	B409	(AH) = 09 (IP) = 0000	(AH) = 09 (IP) = 0002
0002	INT 21H	CD21	(IP) = 0002	(IP) = 0004
0004	RET	C3	(SP) = 0012 (IP) = 0004 Stack +0 = 001A Stack +2 = 0000 Stack +4 = 19F5	(SP) = 0014 (IP) = 001A Stack +0 = 0000 Stack +2 = 19F5 Stack +4 = 0000
001A	RET	CB	(SP) = 0014 (CS) = 1A0A (IP) = 001A Stack +0 = 0000 Stack +2 =	(SP) = 0018 (CS) = 19F5 (IP) = 0000 Stack +0 = 0000 Stack +2 =

			19F5	0000
0000	INT 20	CD20	(IP) = 0000	(IP) = 0005

### **Выводы.**

В ходе выполнения работы мы ознакомились с трансляцией, линковкой и выполнением двух программ на языке Ассемблер. Также обе программы были успешно запущены и протестированы.

## 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.1303 – Самохин К.А.',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
```

```
; HELLO1.ASM - гİа®йГ-- п ŸГабЁп гзГŸ-®© İа®Ja
¬¬Л « Ÿ.а Ÿ. N1
;      İ® ¤ЁбжЁİ«Ё-Г "ЪаеЁвГЄвга Є®¬İ
movГa "
; *****
*****
; К §- зГ-ЁГ: Ца®Ja ¬¬ д®а¬ЁагГв Ё ŸлŸ®¤Ёв -
нСа - İаЁŸГвбвŸЁГ
;      İ®«м§®Ÿ вГ«п б İ®¬®ймо дГ-ЄжЁЁ „Һ
“ „лŸ®¤ бва®ЄЁ"
;      (-®¬Га 09 İаГалŸ -ЁГ 21h), Є®в®а
п:
;      - ®ŸГбİГзЁŸ Гв ŸлŸ®¤ - нСа - бв
а®ЄЁ бЁ¬Ÿ®«®Ÿ,
;      § Є -зЁŸ ойГ©бп §- Є®¬ "$";
;      - ваГŸГГв § ¤ -Ёп Ÿ аГЈЁбваГ ah
-®¬Га дГ-ЄжЁЁ=09h,
;      Ÿ аГЈЁбваГ dx - б¬ГйГ-Ёп
¤аГб ŸлŸ®¤Ё¬®©
;      бва®ЄЁ;
;      - Ёбİ®«м§ГГв аГЈЁбва ах Ё -Г
б®еа -пГв ГЈ®
;      б®¤ГaİЁ¬®Г.
; *****
*****
```

```

                                DOSSEG
                                ; ħ ħ -ĖĲ ĲĲĲ-Ĳ-B@Ÿ Ĳ@Ĳ „Ħ“
                                .MODEL SMALL
                                ; Ħ@ĲĲ«M Ĳ Ĳ-ĲBĖ-SMALL(Ħ « Ĳ)
                                .STACK 100h
                                ; ĦBŸĲĲBĖ Ĳ@Ĳ ‘BĲ€ 256 Ÿ ©B
                                .DATA
                                ; Ĳ 3 «@ ĲĲĲ-Ĳ-B ħ --Ĳe
0000      Greeting LABEL BYTE
                                ; 'Ĳ€ĲB ĲaĖŸĲBĲBŸĲĲB
0000 82 A0 E1 20 AF E0      DB ', Ĳ ĲaĖŸĲBĲBŸĲĲB ĲB.Ja.7303 - €Ÿ -@Ÿ €€
                                .',13,10','$'

                                A8 A2 A5 E2 E1 E2
                                A2 E3 A5 E2 20 E1
                                E2 2E A3 E0 2E 37
                                33 30 33 20 2D 20
                                88 A2 A0 AD AE A2
                                20 88 2E 88 2E 0D
                                0A 24

                                .CODE                                ; Ĳ 3 «
                                @ ĲĲĲ-Ĳ-B €@Ĳ
0000 B8 ---- R      mov ax, @data                                ; ħ Ja
                                ĲŸ€ Ÿ DS ĲaĲĲ - 3 «
0003 8E D8      mov ds, ax                                ; ĲĲĲ-
                                Ĳ-B ħ --Ĳe
0005 BA 0000 R      mov dx, OFFSET Greeting                                ; ħ Ja
                                ĲŸ€ Ÿ dx Ĳ-ĲŸĲ-ĖĲ
Microsoft (R) Macro Assembler Version 5.10      9/21/22 22:32:25

```

```

                                ; ґаГб
                                ЪГЄбВ ЁаЁЎГббЎЁП
0008      DisplayGreeting:
0008 B4 09      mov ah, 9      ; # дГ
                                -ЄжЁЁ „Ң‘ Ёз вЁ бва®ЄЁ
000A CD 21      int 21h      ; ЎлЎ®
                                - нЄа - ЁаЁЎГббЎЁП
000C B4 4C      mov ah, 4ch      ; # дГ
                                -ЄжЁЁ „Ң‘ § ЎГаиГ-ЁП Ёа®Ја ¬¬л
000E CD 21      int 21h      ; § ЎГа
                                иГ-ЁГ Ёа®Ја ¬¬л Ё ўле®ў ў „Ң‘
                                END

```

Microsoft (R) Macro Assembler Version 5.10

9/21/22 22:32:25

## Symbols-1

## Segments and Groups:

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

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 1303 - \$'

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

```

; HELLO2 - PJC PμP±PSP°CЦ PïCЪPsPiCЪP°PjPjP°
N2
P»P°P±.CЪP°P±.#1 PïPs PrPëCГC†PëPïP»PëPSPμ "P
ђCЪC...PëC,PμPëC,CГCЪP°
PePsPjPïCЪCЪC,PμCЪP°"
; PұCЪPsPiCЪP°PjPjP° PëCГPïPsP»CЪP·CГP
μC, PïCЪPsC†PμPrCГCЪCГ PrP»CЦ PïPμC‡P°C,Pë
CГC,
CЪPsPePë
;
; PŷP•PљPŶPŷ PұP PhP“P PhPњPњP«
= 0024 EOFLine EQU '$' ;
PhPïCЪPμPrPμP»PμPSP
ëPμ CГPëPjPIPsP»CЪPSPsPN₂ PePsPSCГC,P°PSC,C<
; "PљPsPSPμC† CГC
,CЪPsPePë"
; PŶC,PμPë PïCЪPsPiCЪP°PjPjC<
ASSUME CS:CODE, SS:AStack
0000 AStack SEGMENT STACK
0000 000C[ DW 12 DUP(') ; PhC,PIPsPrPëC,CГC

```

```

        ] 12 CÍP»PsPI PïP°PjCÏC,Pë
0021

        ]

0018        AStack    ENDS

        ; P”P°PSPSC<Pµ PïCṪPsPiCṪP°PjPjC<

0000        DATA    SEGMENT

        ; P”PëCṪPµPeC,PëPIC< PsPïPëCÍP°PSPëCÏ 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 6EGREETING DB 'Student from 4350 - $'
        74 20 66 72 6F 6D
        20 34 33 35 30 20
        2D 20 24
0025        DATA    ENDS

        ; PḷPsPr PïCṪPsPiCṪP°PjPjC<

0000        CODE     SEGMENT

        ;          PḷCṪPsC†PµPrCfCṪP°          PïPµC‡P°C,Pë
CÍC,CṪPsPePë

0000        WriteMsg PROC NEAR

```

```

0000 B4 09                mov  AH,9
0002 CD 21                int  21h ; P'C<P·PsPI C,,CfPSPeC†PëP
                                ë DOS PïPs PïCṪPμCṪC<PIP°PSPëCṪ
0004 C3                    ret
0005                      WriteMsg ENDP

```

```

; P“PsP»PsPIPSP°CṪ PïCṪPsC†PμPrCfCṪP°

```

Microsoft (R) Macro Assembler Version 5.10 9/21/22 22:33:14

Page 1-2

```

0005                      Main  PROC FAR
0005 1E                    push DS    ;\ PŸPsC...CṪP°PSPμPSPë
                                Pμ P°PrCṪPμCṪP° PSP°C‡P°P»P° PSP PI CṪC,PμPePμ
0006 2B C0                sub  AX,AX  ;> PrP»CṪ PïPsCṪP»PμP
                                rCfCṪC%oPμPiPs PIPsCṪCṪC,P°PSPsPIP»PμPSPëCṪ
PïPs
0008 50                    push AX    ;/ PePsPjP°PSPrPμ ret
                                , P·P°PIPμCṪC€P°CṪC%oPμPNö
PïCṪPsC†PμPrCfCṪCf.
0009 B8 ---- R            mov  AX,DATA    ; P—P°PiCṪC
                                íP·PeP° CṪPμPiPjPμPSC,PSPsPiPs
000C 8E D8                mov  DS,AX      ; CṪPμPiPëC
                                ÍC,CṪP° PrP°PSPSC<C....
000E BA 0000 R            mov  DX, OFFSET HELLO ; P'C<PIPsP
                                r PSP° CṪPeCṪP°PS PïPμCṪPIPsPNö
0011 E8 0000 R            call WriteMsg    ; CṪC,CṪPsP
                                ePë PïCṪPëPIPμC,CṪC,PIPëCṪ.
0014 BA 0010 R            mov  DX, OFFSET GREETING ; P'C<PIPsP

```

```

        r PSP° CkPεCтP°PS PIC,PsCтPsPNε
0017 E8 0000 R          call WriteMsg          ; CГC,CтPsP
        εPε PиCтPεPIPμC,CГC,PIPεCЦ.
001A CB                ret                    ; P'C<C...PsP
        r PI DOS PиPs PePsPjP°PSPrPμ,
                                ; PSP°C...PsP
        rCЦC%oPμPNεCГCЦ PI 1-PsPj CГP»PsPIPμ PSP.
001B                   Main    ENDP
001B                   CODE    ENDS
                                END Main

```

Microsoft (R) Macro Assembler Version 5.10 9/21/22 22:33:14

# Symbols-1

## Segments and Groups:

N a m e	Length	AlignCombine	Class
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

47990 + 459267 Bytes symbol space free

0 Warning Errors

0 Severe Errors