

**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. 1381

Мелькумянц Д.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

### **Цель работы.**

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

### **Задание.**

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, по каждой из которых необходимо представить протокол с фиксацией всех выполняемых действий и полученных результатов, и подписать его у преподавателя.

### **Ход работы.**

1. Просмотрена программа `hello1.asm`, которая формирует и выводит на экран приветствие пользователя с помощью функции ОС MSDOS, вызываемой через прерывание с номером 21H.

2. Разобраться в структуре и реализации каждого сегмента программы. Непонятные фрагменты прояснить у преподавателя. Строку-приветствие преобразовать в соответствии со своими личными данными.

3. Загружен файл `hello1.asm` из каталога Задания в каталог Masm.

4. Протранслирована программа с помощью строки `> masm hello1.asm` с созданием объектного файла и файла диагностических сообщений (файла листинга).

5. Скомпонован загрузочный модуль с помощью строки `> link hello1.obj` с созданием карты памяти и исполняемого файла `hello1.exe`.

6. Выполнена программа в автоматическом режиме путем набора строки `> hello1.exe`

7. Запущено выполнение программы под управлением отладчика с помощью команды

> afdpro hello1.ex

hello1.asm

Адрес команды	Символический код команды	16-ричный код команды	Изменяемые данные	
			до	после
0010	MOV AX, 1A07	B8071A	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 DX = 1A07	IP = 0018 DX = 0000
0018	MOV AH, 09	B409	AX = 1A07 IP = 0018	AX = 0907 IP = 001A
001A	INT 21	CD21	IP = 001A	IP = 001C
001C	MOV AH, 4C	B44C	AX = 0907 IP = 001C	AX = 4C07 IP = 001E
001E	INT 21	CD21	AX = 4C07 DS = 1A07 IP = 001E CX=004E	AX = 0000 DS = 19F5 IP = 0010 CX=0000

hello2.asm

Адрес команды	Символический код команды	16-ричный код команды	Изменяемые данные	
			до	после
0005	PUSH DS	1E	SP = 0018 IP = 0005 DS = 19F5 STACK(+0) = 0000	SP = 0016 IP = 0006 DS = 19F5 STACK(+0) = 19F5
0006	SUB AX,AX	2BC0	IP = 0006 AX=0000	IP = 0008 AX=0000
0008	PUSH AX	50	SP = 0016 IP = 0008 AX = 0000 STACK(+0) = 19F5 STACK(+2) = 0000	SP = 0014 IP = 0009 AX = 0000 STACK(+0) = 0000 STACK(+2) = 19F5
0009	MOV AX,1A07	B8071A	IP = 0009 AX = 0000	IP = 000C AX = 1A07

000C	MOV DS,AX	8ED8	DS = 19F5 IP = 000C AX = 1A07	DS = 1A07 IP = 000E AX = 1A07
000E	MOV DX,0000	BA0000	IP = 000E DX = 0000	IP = 0011 DX = 0000
0011	CALL 0000	E9ECFF	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, 09	B409	AX = 1A07 IP = 0000	AX = 0907 IP = 0002
0002	INT 21	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	BA1000	DX = 0000 IP = 0014	DX = 0010 IP = 0017
0017	CALL 0000	3836FF	IP = 0017 SP = 0014 STACK(+0) = 0000 STACK(+2) = 19F5 STACK (+4) = 0000	IP = 0000 SP = 0012 STACK(+0) = 001A STACK(+2) = 0000 STACK (+4) = 19F5
0000	MOV AH, 09	B409	IP = 0000 AX = 0907	IP = 0002 AX = 0907
0002	INT 21	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 FAR	CB	CS = 1A0A SP = 0014 IP = 001A STACK(+2) = 19F5	CS = 19F5 SP = 0018 IP = 0000 STACK(+2) = 0000
0000	INT 20	CD20	AX = 0907 DX = 0010 CS = 19F5 DS = 1A07 IP = 0000 CX = 007B	AX = 0000 DX = 0000 CS = 1A0A DS = 19F5 IP = 0005 CX = 0000

**Выводы.**

В ходе выполнения лабораторной работы были получены основные навыки программирования на ассемблере.

## Приложение А. Код программ.

Имя 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.1381 - Новак П.И.',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
```

Имя 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  1381  –  Novak  Polina$'
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. Листинг успешной трансляции  
программами.**

Имя файла: hello1.lst

#Microsoft (R) Macro Assembler Version 5.10

9/7/22 20:37:33

Page 1-1

```
PIРμСѢСГРё ; HELLO1.ASM - CѓPіCѢPsC%PμPSPSP°CЦ
CЦ CѓCѓPμP±PSPsPNё PіCѢPsPіCѢP°PjPjC<
P»P°P±.CѢP °P±. N1
; PіPs PrPёCѓCѓPёPіP»PёPSPμ "PђCѢ
C...PёC,PμPёC,CѓCѢP° PёPsPjPіCђCѢC,PμCѢP°"
; *****
*****
; PќP°P·PSP°CѓPμPSPёPμ: PђCѢPsPіCѢP°PjPjP° C,,Ps
CѢPjPёCѢCѓPμC, Pё PIC<PIPsPrPёC, PSP° CќPёCѢP°P
S PіCѢPёPIРμC,CѓC,PіPёPμ
; PіPsP»CђP·PsPIP°C,PμP»CЦ Cѓ PіPsP
jPsC%CђCѢ C,,CѓPSPёCѓPёPё P”PђPŸ "P’C<PIPsPr
CѓC ,CѢPsPёPё"
; (PSPsPjPμCѢ 09 PіCѢPμCѢC<PIP°PSPё
Pμ 21h), PёPsC,PсCѢP°CЦ:
; - PsP±PμCѓPіPμCѓPёPIP°PμC, PIC<P
IPsPr PSP° CќPёCѢP°PS CѓC,CѢPsPёPё CѓPёPjPIPsP»
PsPI,
; P·P°PёP°PSCѓPёPIP°CѢC%PμPNёCѓCЦ
P·PSP°PёPsPj "$";
; - C,CѢPμP±CѓPμC, P·P°PrP°PSPёCЦ
PI CѢPμPіPёCѓC,CѢPμ ah PSPsPjPμCѢP° C,,CѓPSPёCѓP
ёPё=09h,
; P° PI CѢPμPіPёCѓC,CѢPμ dx - C
ЃPjPμC%PμPSPёCЦ P°PrCѢPμCѓP° PIC<PIPsPrPёPjPs
PNё
; CѓC,CѢPsPёPё;
; - PёCѓPіPsP»CђP·CѓPμC, CѢPμPіPёC
ЃC,CѢ ax Pё PSPμ CѓPsC...CѢP°PSCЦPμC, PμPіPs
; CѓPsPrPμCѢPђPёPjPsPμ.
; *****
```

\*\*\*\*\*

```
DOSSEG
; P—P°PrP°PSPëPµ CÍPµPiPjPµPSC,PσPI PïPsPr P”Ph
PŸ
.MODEL SMALL
; PµPsPrPµP»CH PïP°PjCÍC,Pë-SMALL(PµP°P»P°CÍ)
.STACK 100h
; PhC,PIPµCÍC,Pë PïPsPr PŸC,PµPε 256 P±P°PNεC,
.DATA
; PκP°C‡P°P»Ps CÍPµPiPjPµPSC,P° PrP°PSPSC<C...
0000 Greeting LABEL BYTE
; PŸPµPεCÍC, PïCṪPëPIPµC,CÍC,PIPëCÍ
0000 D0 92 D0 B0 D1 81 DB 'P’P°CÍ PïCṪPëPIPµC,CÍC,PICÍPµC,
CÍC,.PiC
Ṫ.1381 - PµPµP»CHPεCÍPjCÍPSC† P”.Ph.',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
31 33 38 31 20 2D
```

#Microsoft (R) Macro Assembler Version 5.10

9/7/22 20:37:33

Page 1-2

```
20 D0 9C D0 B5 D0
BB D1 8C D0 BA D1
83 D0 BC D1 8F D0
BD D1 86 20 D0 94
2E D0 90 2E 0D 0A
24
.CODE ; PκP°C
‡P°P»Ps CÍPµPiPjPµPSC,P° PεPsPrP°
0000 B8 ---- R mov ax, @data ; P—P°
PïCṪCÍP·PεP° PI DS P°PrCṪPµCÍP° 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°
PïCṪCÍP·PεP° PI dx CÍPjPµC%PµPSPëCÍ
; P°PrC
ṪPµCÍP° C,PµPεCÍC,P° PïCṪPëPIPµC,CÍC,PIPëCÍ
0008 DisplayGreeting:
```

0008 B4 09	mov ah, 9 ; # C,, CfPSPeC†PëPë P”PhPŸ PiPμC‡P°C,Pë CÍC,CṪPsPePë
000A CD 21	int 21h ; PIC<P IPsPr PSP° CÍPeCṪP°PS PiCṪPëPIPμC,CÍC,PIPëCŪ
000C B4 4C	mov ah, 4ch ; # C,, CfPSPeC†PëPë P”PhPŸ P·P°PIPμCṪC€PμPSPëCŪ
PiCṪPs	PiCṪP°PjPjC<
000E CD 21	int 21h ; P·P°P IPμCṪC€PμPSPëPμ PiCṪPsPiCṪP°PjPjC< Pë PIC<C...PsP r PI P”PhPŸ END

## Symbols-1

## Segments and Groups:

N a m e	Length	Align	Combine	Class
DGROUP .....	GROUP			
_DATA .....	004F 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

Имя файла: hello2.lst

```

; HELLO2 - PJC PμP±PSP°C PīCṪPsPīCṪP°PjPjP° N2
P»P°P±.CṪP°P±.#1 PīPs PrPēCṪC†PēPīP»PēPSPμ "P
ḡCṪC...PēC,PμPēC,CṪCṪP° PēPsPjPīCḤCṪC,PμCṪP°"
; PṫCṪPsPīCṪP°PjPjP° PēCṪPīPsP»CḤP·CṪP
μC, PīCṪPsC†PμPrCṪCṪCṪ PrP»CṪ PīPμCṪP°C,Pē CṪC,
CṪPsPēPē
;
; PŷP•PḷPŶPŷ PṫP PhP“P PhPḤPḤP«

= 0024 EOFLine EQU '$' ; PhPīCṪPμPrPμP»PμPSP
ēPμ CṪPēPjPIPsP»CḤPSPsPNḡ PēPsPSCṪC,P°PSC,CṪ
; "PḷPsPSPμC† CṪC
,CṪPsPēPē"

; PŶC,PμPē PīCṪPsPīCṪP°PjPjCṪ

ASSUME CS:CODE, SS:AStack

0000 AStack SEGMENT STACK
0000 000C[ DW 12 DUP('') ; PhC,PIPsPrPēC,CṪC
Ṫ 12 CṪP»PsPI PīP°PjCṪC,Pē
0021 ]

0018 AStack ENDS

; P”P°PSPSCṪPμ PīCṪPsPīCṪP°PjPjCṪ

0000 DATA SEGMENT

; P”PēCṪPμPēC,PēPICṪ PsPīPēCṪP°PSPēCṪ PrP°PSPS
CṪC...

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 – Novak Polina$'
74 20 66 72 6F 6D
20 34 33 35 30 20
2D 20 24

```

0025 DATA ENDS

; PЉPsPr PiCЉPsPiCЉP°PjPjC<

0000 CODE SEGMENT

; PЎCЉPsC†PμPrCfCЉP° PiPμC‡P°C,Pë CЉC,CЉPsPePë

0000 WriteMsg PROC NEAR

0000 B4 09 mov AH,9

0002 CD 21 int 21h ; P’C<P·PsPI C,,CfPSPeC†PëP

ë DOS PiPs PiCЉPμCЉC<PIP°PSPëCЉ

0004 C3 ret

0005 WriteMsg ENDP

; P“PsP»PsPIPSP°CЉ PiCЉPsC†PμPrCfCЉP°

#Microsoft (R) Macro Assembler Version 5.10

9/7/22 20:50:36

Page 1-2

0005 Main PROC FAR

0005 1E push DS ;\ PŸPsC...CЉP°PSPμPSPë

Pμ P°PrCЉPμCЉP° PSP°C‡P°P»P° PSP PI CЉC,PμPePμ

0006 2B C0 sub AX,AX ; > PrP»CЉ PiPsCЉP»PμP

rCfCЉC%PμPiPs PIPsCЉCЉC,P°PSPsPIP»PμPSPëCЉ

PiPs

0008 50 push AX ;/ PePsPjP°PSPrPμ ret

, P·P°PIPμCЉC€P°CЉC%PμPNë PiCЉPsC†PμPrCfCЉCf.

0009 B8 ---- R mov AX,DATA ; P—P°PiCЉC

řP·PeP° CЉPμPiPjPμPSC,PSPsPiPs

000C 8E D8 mov DS,AX ; CЉPμPiPëC

ЉC,CЉP° PrP°PSPSC<C....

000E BA 0000 R mov DX, OFFSET HELLO ; P’C<PIPsP

r PSP° CЉPeCЉP°PS PiPμCЉPIPsPNë

0011 E8 0000 R call WriteMsg ; CЉC,CЉPsP

ëPë PiCЉPëPIPμC,CЉC,PIPëCЉ.

0014 BA 0010 R mov DX, OFFSET GREETING ; P’C<PIPsP

r PSP° CЉPeCЉP°PS PIC,PsCЉPsPNë

0017 E8 0000 R call WriteMsg ; CЉC,CЉPsP

ëPë PiCЉPëPIPμC,CЉC,PIPëCЉ.

001A CB ret ; P’C<C...PsP

r PI DOS PiPs PePsPjP°PSPrPμ,

; PSP°C...PsP

rCЉC%PμPNëCЉCЉ PI 1-PsPj CЉP»PsPIPμ PSP.

001B Main ENDP

001B CODE ENDS

END Main

## Symbols-1

## Segments and Groups:

N a m e	Length	Align	Combine	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

47986 + 459271 Bytes symbol space free

0 Warning Errors

0 Severe Errors