

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МОЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №1**  
**по дисциплине «Организация ЭВМ и систем»**  
**Тема: Освоение трансляции, выполнения и отладки программ на языке**  
**Ассемблера**

Студент гр.1381

\_\_\_\_\_

Луценко Д. А.

Преподаватель

\_\_\_\_\_

Ефремов М.А.

Санкт-Петербург

2022

## **Цель работы.**

Познакомиться с работой простейших программ на языке Ассемблер. Выполнить их трансляцию, компоновку и отладку.

## **Задание.**

### *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. 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 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 созданием карты памяти и исполняемого файла `hello1.exe`.

6. Выполнить программу в автоматическом режиме и убедиться в корректности ее работы и зафиксировать результат выполнения в протоколе.

7. Запустить выполнение программы под управлением отладчика. 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.

## *Часть 2*

Выполнить пункты 1 - 7 части 1 настоящего задания применительно к программе hello2.asm, приведенной в каталоге Задания, которая выводит на экран приветствие пользователя с помощью процедуры WriteMsg, а также использует полное определение сегментов. Сравнить результаты прогона под управлением отладчика программ hello1 и hello2 и объяснить различия в размещении сегментов.

### **Выполнение работы.**

Начальные значения регистров: (AX) = 0000, (BX) = 0000, (CX) = 0067, (DX) = 0000, (CS) = 1A05, (DS) = 19F5, (ES) = 19F5, (SS) = 1A0C.

Результаты пошагового выполнения представлены в таблице 1.

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров	
			До выполнения	После выполнения
0010	MOV AX, 1A07	B8071A	(AX) = 0000 (IP) = 0010	(AX) = 1A07 (IP) = 0013

0013	MOV DS, AX	8ED8	(DS) = 19F5 (IP) = 0013	(DS) = 1A07 (IP) = 0015
0015	MOV DX, 0000	BA0000	(IP) = 0015	(IP) = 0018
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 (CX) = 0067 (DS) = 1A07 (IP) = 001E	(AX) = 0000 (CX) = 0000 (DS) = 19F5 (IP) = 0010

## Часть 2:

1. Для файла *hello2.asm* были проделаны все те же шаги, что и для предыдущего.

2. Начальные значения регистров: (AX) = 0000, (BX) = 0000, (CX) = 006B, (DX) = 0000, (CS) = 1A0A, (DS) = 19F5, (ES) = 19F5, (SS) = 1A05. Результаты пошагового выполнения представлены в таблице 2.

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	(AX) = 0000 (IP) = 0006	(AX) = 0000 (IP) = 0008
0008	PUSH AX	50	(SP) = 0016 (IP) = 0008 Stack: +0 19F5 +2 0000	(SP) = 0014 (IP) = 0009 Stack: +0 0000 +2 19F5
0009	MOV AX, 1A07	B8071A	(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	(IP) = 000E	(IP) = 0011
0011	CALL 0000	E8ECFF	(SP) = 0014 (IP) = 0011 Stack: +0 0000 +2 19F5 +4 0000	(SP) = 0012 (IP) = 0000 Stack: +0 0014 +2 0000 +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 +2 0000 +4 19F5	(SP) = 0014 (IP) = 0014 Stack: +0 0000 +2 19F5 +4 0000
0014	MOV DX, 0010	BA1000	(IP) = 0014	(IP) = 0017
0017	CALL 0000	E8E6FF	(SP) = 0014 (IP) = 0017	(SP) = 0012 (IP) = 0000

			Stack: +0 0000 +2 19F5 +4 0000	Stack: +0 001A +2 0000 +4 19F5
0000	MOV AH, 09	B409	(IP) = 0000	(IP) = 0002
0002	INT 21	CD21	(IP) = 0002	(IP) = 0004
0004	RET	C3	(SP) = 0012 (IP) = 0004 Stack: +0 001A +2 0000 +4 19F5	(SP) = 0014 (IP) = 001A Stack: +0 0000 +2 19F5 +4 0000
001A	RET Far	CB	(SP) = 0014 (CS)=1A0A (IP) = 001A Stack: +2 19F5	(SP) = 0018 (CS)=19F5 (IP) = 0000 Stack: +2 0000
0000	INT 20	CD20	(AX)=0907 (CX)=006B (CS) = 19F5 (DS) = 1A07 (IP) = 0000	(AX) = 0000 (CX) = 0000 (CS) = 1A0A (DS) = 19F5 (IP) = 0005

Таблица 2

### Выводы.

В ходе выполнения лабораторной работы были рассмотрены процессы трансляции и отладки на языке Assembler.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД 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 'Вас приветствует ст.гр.1381 - Луценко Д.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
```

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 1381 - $'
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

## ЛИСТИНГОВЫЕ ФАЙЛЫ

Название файла: *HELLO1.LST*

Microsoft (R) Macro Assembler Version 5.10  
23:42:26

9/14/22

Page 1-1

```
; HELLO1.ASM - CíPíCtPsC%PµPSPSP°Cµ PIPµCtCíPë
Cµ CíCtPµP±PSPsPN° PíCtPsPíCtP°PjPjC< P»P°P±.CtP
°P±. N1
;
; PíPs PrPëCíCtPëPíP»PëPSPµ "PjCt
C...PëC, PµPeC, CíCtP° PePsPjPíCtCtC, PµCtP°"
; *****
*****
; PkP°P·PSP°CtPµPSPëPµ: PµCtPsPíCtP°PjPjP° C„Ps
CtPjPëCtCíPµC, Pë PIC<PIPsPrPëC, PSP° CkPeCtP°P
S PíCtPëPIPµC, CíC, PIPëPµ
;
; PíPsP»CtP·PsPIP°C, PµP»Cµ Cí PíPsP
jPsC%CtCt C„CíPSPeCtPëPë P”PhPŸ "P’C<PIPsPr CíC
, CtPsPePë"
;
; (PSPsPjPµCt 09 PíCtPµCtC< PIP°PSPë
Pµ 21h), PePsC, PsCtP°Cµ:
;
; - PsP±PµCíPíPµCtPëPIP°PµC, PIC<P
IPsPr PSP° CkPeCtP°PS CíC, CtPsPePë CíPëPjPIPsP»
PsPI,
;
; P·P°PeP°PSCtPëPIP°CtC%PµPN°CíCµ
P·PSP°PePsPj "$";
;
; - C, CtPµP±CíPµC, P·P°PrP°PSPëCµ
PI CtPµPíPëCíC, CtPµ ah PSPsPjPµCtP° C„CíPSPeCtP
ëPë=09h,
;
; P° PI CtPµPíPëCíC, CtPµ dx - C
íPjPµC%PµPSPëCµ P°PrCtPµCíP° PIC<PIPsPrPëPjPs
PN°
;
; CíC, CtPsPePë;
;
; - PëCíPíPsP»CtP·CíPµC, CtPµPíPëC
íC, Ct ax Pë PSPµ CíPsC...CtP°PSCµPµC, PµPíPs
;
; CíPsPrPµCtPµPëPjPsPµ.
; *****
*****
```

```
DOSSEG
; P-P°PrP°PSPëPµ CíPµPíPjPµPSC, PsPI PíPsPr P”Ph
PŸ
.MODEL SMALL
; PµPsPrPµP»Ct PíP°PjCµC, Pë-SMALL (PµP°P»P°Cµ)
.STACK 100h
; PhC, PIPµCíC, Pë PíPsPr PŸC, PµPe 256 P±P°PN°C,
.DATA
; PkP°CtP°P»Ps CíPµPíPjPµPSC, P° PrP°PSPSC<C...
Greeting LABEL BYTE
```

0000

```

; PŸPµPeCÍC, PiCŤPĚPIPµC,CÍC,PIPĚCŮ
0000 D0 92 D0 B0 D1 81 DB 'P'P°CÍ PiCŤPĚPIPµC,CÍC,PICÍPµC, CÍC,.PiC
      7.1381 - P>CÍC†PµPSPePs P".PŤ.',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          9/14/22
23:42:26

```

Page 1-2

```

      20 D0 9B D1 83 D1
      86 D0 B5 D0 BD D0
      BA D0 BE 20 D0 94
      2E D0 90 2E 0D 0A
      24
      .CODE
      ‡P°P»Ps CÍPµPiPjPµPSC,P° PePsPrP°          ; PkP°C
0000 B8 ---- R      mov ax, @data          ; P-P°
      PiCŤCÍP·PeP° PI DS P°PrCŤPµCÍP° 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°
      PiCŤCÍP·PeP° PI dx CÍPjPµC%PµPSPĚCŮ
      ; P°PrC
      7PµCÍP° C,PµPeCÍC,P° PiCŤPĚPIPµC,CÍC,PIPĚCŮ
0008 DisplayGreeting:
0008 B4 09          mov ah, 9          ; # C,,
      CÍPSPeC†PĚPĚ P"PhPŸ PiPµC†P°C,PĚ CÍC,CŤPsPePĚ
000A CD 21          int 21h          ; PIC<P
      IPsPr PSP° CÍPeCŤP°PS PiCŤPĚPIPµC,CÍC,PIPĚCŮ
000C B4 4C          mov ah, 4ch          ; # C,,
      CÍPSPeC†PĚPĚ P"PhPŸ P·P°PIPµCŤCĚPµPSPĚCŮ PiCŤPs
      PiCŤP°PjPjC<
000E CD 21          int 21h          ; P·P°P
      IPµCŤCĚPµPSPĚPµ PiCŤPsPiCŤP°PjPjC< PĚ PIC<C...PsP
      r PI P"PhPŸ
      END
Microsoft (R) Macro Assembler Version 5.10          9/14/22
23:42:26

```

Symbols-1

# Segments and Groups:

	N a m e	Length	Align	Combine	Class
DGROUP	. . . . .	GROUP			
_DATA	. . . . .	0049	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

## Название файла: *HELLO2.LST*

Microsoft (R) Macro Assembler Version 5.10

9/14/22

23:43:52

Page 1-1

```
; HELLO2 - PJC+PμP+PSP°CЇ PìCЪPsPìCЪP°PjPjP° N2
P»P°P±.CЪP°P±.#1 PìPs PrPëCíC†PëPìP»PëPSPμ "P
ЇCЪC...PëC,PμPeC,CíCЪP° PePsPjPìCЪCЪC,PμCЪP°"
; PμCЪPsPìCЪP°PjPjP° PëCíPìPsP»CЪP·CíP
μC, PìCЪPsC†PμPrCíCЪCí PrP»CЇ PìPμC†P°C,Pë CíC,
CЪPsPePë
;
; PŸP•PљPŸPŸ PμP PђP"P PђPђPђP«
```

```
= 0024 EOFLine EQU '$' ; PђPìCЪPμPrPμP»PμPSP
ëPμ CíPëPjPIPsP»CЪPSPsPN PePsPSCíC,P°PSC,C<
; "PљPsPSPμC† CíC
,CЪPsPePë"
```

```
; PŸC,PμPe PìCЪPsPìCЪP°PjPjC<
```

```
ASSUME CS:CODE, SS:AStack
```

```
0000 AStack SEGMENT STACK
0000 000C[ DW 12 DUP('!') ; PђC,PIPsPrPëC,CíC
Ї 12 CíP»PsPI PìP°PjCЇC,Pë
0021
]
0018 AStack ENDS
```

; P°PSPSC<Pµ PiCßPsPiCßP°PjPjC<

0000 DATA SEGMENT

; P°PëCßPµPeC,PëPIC< PsPiPëCíP°PSPëCµ 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 31 33 38 31 20  
2D 20 24

0025 DATA ENDS

; PµPsPr PiCßPsPiCßP°PjPjC<

0000 CODE SEGMENT

; PµCßPsC†PµPrCíCßP° PiPµC†P°C,Pë CíC,CßPsPePë

0000 WriteMsg PROC NEAR

0000 B4 09 mov AH,9

0002 CD 21 int 21h ; P'C<P·PsPI C,,CíPSPeC†PëP

ë DOS PiPs PiCßPµCßC<PIP°PSPëCß

0004 C3 ret

0005 WriteMsg ENDP

; P°PsP»PsPIPSP°Cµ PiCßPsC†PµPrCíCßP°

\_\_Microsoft (R) Macro Assembler Version 5.10

9/14/22

23:43:52

Page 1-2

0005 Main PROC FAR

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

Pµ P°PrCßPµCíP° PSP°C†P°P»P° PSP PI CíC,PµPePµ

0006 2B C0 sub AX,AX ; > PrP»Cµ PiPsCíP»PµP

rCíCßC%PµPiPs PIPsCíCíC,P°PSPsPIP»PµPSPëCµ PiPs

0008 50 push AX ;/ PePsPjP°PSPPrPµ ret

, P·P°PIPµCßCëP°CßC%PµPN° PiCßPsC†PµPrCíCßCí.

0009 B8 ---- R mov AX,DATA ; P-P°PiCßC

íP·PeP° CíPµPiPjPµPSC,PSPsPiPs

000C 8E D8 mov DS,AX ; CßPµPiPëC

íC,CßP° PrP°PSPSC<C...

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

r PSP° CíPeCßP°PS PiPµCßPIPsPN°

0011 E8 0000 R call WriteMsg ; CíC,CßPsP

ePë PiCßPëPIPµC,CíC,PIPëCµ.

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

r PSP° CíPeCßP°PS PIC,PsCßPsPN°

0017 E8 0000 R call WriteMsg ; CíC,CßPsP

ePë PiCßPëPIPµC,CíC,PIPëCµ.

001A CB ret ; P'C<C...PsP

r PI DOS PiPs PePsPjP°PSPPrPµ,

; PSP°C...PsP

rCµC%PµPN°CíCµ PI 1-PsPj CíP»PsPIPµ PSP.

```

001B          Main      ENDP
001B          CODE      ENDS
END Main
Microsoft (R) Macro Assembler Version 5.10
23:43:52
Symbols-1

```

9/14/22

#### 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

```