

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ЭЛЕК-**  
**ТРОТЕХНИЧЕСКИЙ 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

\_\_\_\_\_

Туркова Д.Н.

Преподаватель

\_\_\_\_\_

Ефремов М.А.

Санкт-Петербург

2022

### **Цель работы.**

Изучить программу на Ассемблере, выводящую строку на экран, её трансляцию, компоновку, отладку.

### **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.

## 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.**

Просмотрена и изучена программа hello1.asm.

Выполнена трансляция программы с созданием объектного файла и файла листинга:

>masm hello1.asm

Выполнена компоновка:

>link hello1.obj

Запущен исполняемый файл hello1.exe.

Далее с помощью отладчика AFDPRO была осуществлена отладка программы и просмотр регистров процессора.

>afdpro.exe hello1.exe

Была проведена аналогичная работа с программой hello2.asm. Выведена строка «Hello Worlds! Student from 4350-»

Результаты отладки обеих программ представлены в табл. 1 и табл. 2.

Файлы листинга см. в приложении А.

Таблица 1 – Результаты отладки программы hello1.exe

Начальные значения сегментных регистров: CS — 1A05, DS - 19F5, ES — 19F5, SS - 1A0C

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0010	MOV AX, 1A07	B8071A	IP = 0010 AX = 0000	IP = 0013 AX = 1A07
0013	MOV DS, AX	8ED8	IP = 0013 AX = 1A07 DS = 19F5	IP = 0015 AX = 1A07 DS = 1A07
0015	MOV DX, 0000	BA0000	IP = 0015 DX = 0000	IP = 0018 DX = 0000
0018	MOV AH, 09	B409	IP = 0018 AX = 1A07	IP = 001A AX = 0907
001A	INT 21	CD21	IP = 001A	IP = 001C
001C	MOV AH, 4C	B44C	IP = 001C AX = 0907	IP = 001E AX = 4C07
001E	INT 21	CD21	IP = 001E	IP = 0010

Таблица 2 – Результаты отладки программы hello2.exe

Начальные значения сегментных 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	IP = 0005 DS = 19F5 SP = 0018	IP = 0006 DS = 19F5 SP = 0016
			Stack +0 0000 +2 0000 +4 0000 +6 0000	Stack +0 19F5 +2 0000 +4 0000 +6 0000

0006	SUB AX, AX	2BC0	IP = 0006 AX = 0000	IP = 0008 AX = 0000
0008	PUSH AX	50	IP = 0008 AX = 0000 SP = 0016 Stack: +0 19F5 +2 0000 +4 0000 +6 0000	IP = 0009 AX = 0000 SP = 0014 Stack: +0 0000 +2 19F5 +4 0000 +6 0000
0009	MOV AX, 1A07	B8071A	IP = 0009 AX = 0000	IP = 000C AX = 1A07
000C	MOV DS, AX	8ED8	IP = 000C AX = 1A07 DS = 19F5	IP = 000E AX = 1A07 DS = 1A07
000E	MOV DX, 0000	BA0000	IP = 000E DX = 0000	IP = 0011 DX = 0000
0011	CALL 0000	E8ECFF	IP = 0011 SP = 0014 Stack +0 0000 +2 19F5 +4 0000 +6 0000	IP = 0000 SP = 0012 Stack +0 0014 +2 0000 +4 19F5 +6 0000
0000	MOV AH, 09	B409	IP = 0000 AX = 1A07	IP = 0002 AX = 0907
0002	INT 21	CD21	IP = 0002	IP = 0004
0004	RET	C3	IP = 0004 SP = 0012	IP = 0014 SP = 0014

			Stack +0 0014 +2 0000 +4 19F5 +6 0000	Stack +0 0000 +2 19F5 +4 0000 +6 0000
0014	MOV DX, 0010	BA 1000	IP = 0014 DX = 0000	IP = 0017 DX = 0010
0017	CALL 0000	E8E6FF	IP = 0017 SP = 0014 Stack +0 0000 +2 19F5 +4 0000 +6 0000	IP = 0000 SP = 0012 Stack +0 001A +2 0000 +4 19F5 +6 0000
0000	MOV AH, 09	B409	IP = 0000 AX = 0907	IP = 0002 AX = 0907
0002	INT 21	CD21	IP = 0002	IP = 0004
0004	RET	C3	IP = 0004 SP = 0012 Stack +0 001A +2 0000 +4 19F5 +6 0000	IP = 001A SP = 0014 Stack +0 0000 +2 19F5 +4 0000 +6 0000
001A	RET Far	CB	IP = 001A SP = 0014 CS = 1A0A Stack +0 0000 +2 19F5 +4 0000 +6 0000	IP = 0000 SP = 0018 CS = 19F5 Stack +0 0000 +2 0000 +4 0000 +6 0000
0000	INT 20	CD20	IP = 0000 AX = 0907 DX = 0010 CS = 19F5	IP = 0005 AX = 0000 DX = 0000 CS = 1A0A

### **Выводы.**

Были изучены два варианта программы на Ассемблере, осуществляющей вывод строки с помощью прерывания 21h, их отладка. Научились запускать программу с помощью Dosbox и приобрели базовые навыки программирования на Assembler.

## ПРИЛОЖЕНИЕ А

Название файла: LST1.LST

Microsoft (R) Macro Assembler Version 5.10  
12:19:44

9/15/22

Page 1-1

```
; HELLO1.ASM - CíPíCtPsC%PµPSPSP°Cµ PíPµCtCíPë
Cµ CíCtPµP±PSPsPN° PíCtPsPíCtP°PjPjC< P»P°P±.CtP
°P±. N1
;
; PíPs PrPëCíCtPëPíP»PëPSPµ "PñCt
C...PëC, PµPeC, CíCtP° PePsPjPíCtCtC, PµCtP°"
; *****
*****
; PkP°P·PSP°CtPµPSPëPµ: PµCtPsPíCtP°PjPjP° C„Ps
CtPjPëCtCíPµC, Pë PíC<PíPsPrPëC, PSP° CkPeCtP°P
S PíCtPëPíPµC, CíC, PíPëPµ
;
; PíPsP»CtP·PsPíP°C, PµP»Cµ Cí PíPsP
jPsC%CtCt C„CíPSPeCtPëPë P"PñPŸ "P'C<PíPsPr CíC
,CtPsPePë"
;
; (PSPsPjPµCt 09 PíCtPµCtC<PíP°PSPë
Pµ 21h), PePsC, PsCtP°Cµ:
;
; - PsP±PµCíPíPµCtPëPíP°PµC, PíC<P
íPsPr PSP° CkPeCtP°PS CíC, CtPsPePë CíPëPjPíPsP»
PsPI,
;
; P·P°PeP°PSCtPëPíP°CtC%PµPN°CíCµ
P·PSP°PePsPj "$";
;
; - C, CtPµP±CíPµC, P·P°PrP°PSPëCµ
Pí CtPµPíPëCíC, CtPµ ah PSPsPjPµCtP° C„CíPSPeCtP
ëPë=09h,
;
; P° Pí CtPµPíPëCíC, CtPµ dx - C
íPjPµC%PµPSPëCµ P°PrCtPµCíP° PíC<PíPsPrPëPjPs
PN°
;
; CíC, CtPsPePë;
;
; - PëCíPíPsP»CtP·CíPµC, CtPµPíPëC
íC, Ct ax Pë PSPµ CíPsC...CtP°PSCµPµC, PµPíPs
;
; CíPsPrPµCtP¶PëPjPsPµ.
; *****
*****
```

```
DOSSEG
; P-P°PrP°PSPëPµ CíPµPíPjPµPSC, PsPI PíPsPr P"Pñ
PŸ
.MODEL SMALL
; PñPsPrPµP»Ct PíP°PjCµC, Pë-SMALL (PñP°P»P°Cµ)
.STACK 100h
; PñC, PíPµCíC, Pë PíPsPr PŸC, PµPe 256 P±P°PN°C,
.DATA
; PkP°CtP°P»Ps CíPµPíPjPµPSC, P° PrP°PSPSC<C...
0000 Greeting LABEL BYTE
; PŸPµPeCíC, PíCtPëPíPµC, CíC, PíPëCµ
0000 D0 92 D0 B0 D1 81 DB 'P'°Cí PíCtPëPíPµC, CíC, PíCíPµC,
CíC,.PíC
B.7303 - P PíP°PSPsPI P 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  
12:19:44

9/15/22

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
0000 B8 ---- R      mov ax, @data            ; P-P°
        PiCbCfP·PeP° PI DS P°PrCbPµCfP° PSP°C+P°P»P°
0003 8E D8          mov ds, ax                ;
CfPµ
        PiPjPµPSC,P° PrP°PSPSC<C...
0005 BA 0000 R      mov dx, OFFSET Greeting    ; P-P°
        PiCbCfP·PeP° PI dx CfPjPµC%PµPSPëCµ
                                ; P°PrC
        BµPµCfP° C,PµPeCfC,P° PiCbPëPIPµC,CfC,PIPëCµ
0008          DisplayGreeting:
0008 B4 09          mov ah, 9                  ; #
C,,
        CfPSPeC†PëPë P"PhPŸ PiPµC†P°C,Pë CfC,CbPsPePë
000A CD 21          int 21h                      ;
PIC<P
        IPsPr PSP° CkPeCbP°PS PiCbPëPIPµC,CfC,PIPëCµ
000C B4 4C          mov ah, 4ch                  ; #
C,,
        CfPSPeC†PëPë P"PhPŸ P·P°PIPµCbCëPµPSPëCµ PiCbPs
        PiCbP°PjPjC<
000E CD 21          int 21h                      ;
P·P°P
        IPµCbCëPµPSPëPµ PiCbPsPiCbP°PjPjC< Pë PIC<C...PsP
        r PI P"PhPŸ
        END
```

Microsoft (R) Macro Assembler Version 5.10  
12:19:44

9/15/22

Symbols-1

#### Segments and Groups:

N a m e	Length	Align	Combine	Class
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

```

47998 + 459262 Bytes symbol space free

0 Warning Errors

0 Severe Errors

## Название файла: LST2.LST

Microsoft (R) Macro Assembler Version 5.10  
12:45:00

9/15/22

Page 1-1

```

; HELLO2 - PJC+PμP+PSP°C PιCБPsPιCБP°PjPjP° N2
; P»P°P±.CБP°P±.#1 PιPs PrPëCíC†PëPιP»PëPSPμ "P
; ħCБC...PëC, PμPeC, CíCБP° PePsPjPιCБCБC, PμCБP°"
; PμCБPsPιCБP°PjPjP° PëCíPιPsP»CБP·CíP
; μC, PιCБPsC†PμPrCíCБCí PrP»C PιPμC†P°C, Pë CíC,
; CБPsPePë
;
; PŸP•PљPŸPŸ PμP PħP"P PħPħPħP«

= 0024 EOFLine EQU '$' ;
PħPιCБPμPrPμP»PμPSP
ëPμ CíPëPjPIPsP»CБPSPsPN° PePsPSCíC, P°PSC, C<
; "PљPsPSEμC† CíC
, CБPsPePë"

; PŸC, PμPe PιCБPsPιCБP°PjPjC<

ASSUME CS:CODE, SS:AStack

0000 AStack SEGMENT STACK
0000 000C[ DW 12 DUP('!') ;
PħC, PιPsPrPëC, CíC
] 12 CíP»PsPI PιP°PjCιC, Pë
0021
]

0018 AStack ENDS

; P"P°PSPSC<Pμ PιCБPsPιCБP°PjPjC<

0000 DATA SEGMENT

; P"PëCБPμPeC, PëPIC< PsPιPëCíP°PSPëC PιP°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 4350 - $'
      74 20 66 72 6F 6D
      20 34 33 35 30 20
      2D 20 24
0025 DATA ENDS

```

```

; PŁPsPr PiCŁPsPiCŁP°PjPjC<

```

```

0000 CODE SEGMENT
; PŁCŁPsCŁPŁPrCŁCŁP° PiPŁCŁP°C,PŁ CŁC,CŁPsPePŁ
0000 WriteMsg PROC NEAR
0000 B4 09 mov AH,9
0002 CD 21 int 21h ; P'C<P·PsPI
C„CŁPSPeCŁPŁP

; DOS PiPs PiCŁPŁCŁC<PIP°PSPŁCŁ
0004 C3 ret
0005 WriteMsg ENDP

```

```

; P`PsP»PsPIPS°CŁ PiCŁPsCŁPŁPrCŁCŁP°

```

```

__Microsoft (R) Macro Assembler Version 5.10 9/15/22
12:45:00

```

Page 1-2

```

0005 Main PROC FAR
0005 1E push DS ;\ PŸPsC...CŁP°PSPŁPSPŁ
PŁ P°PrCŁPŁCŁP° PSP°CŁP°P»P° PSP PI CŁC,PŁPePŁ
sub AX,AX ; > PrP»CŁ

rCŁCŁC%PŁPiPs PIPsCŁCŁC,P°PSPsPIP»PŁPSPŁCŁ PiPs
0008 50 push AX ;/ PePsPjP°PSPPrPŁ ret
, P·P°PIPŁCŁCŁP°CŁC%PŁPN° PiCŁPsCŁPŁPrCŁCŁCŁ.
0009 B8 ---- R mov AX,DATA ; P-P°PiCŁC
řP·PeP° CŁPŁPiPjPŁPSC,PSPsPiPs
000C 8E D8 mov DS,AX ;
CŁPŁPiPŁC

íC,CŁP° PrP°PSPSC<C...
000E BA 0000 R mov DX, OFFSET HELLO ; P'C<PIPsP
r PSP° CŁPeCŁP°PS PiPŁCŁPIPsPN°
0011 E8 0000 R call WriteMsg ; CŁC,CŁPsP
ePŁ PiCŁPŁPIPŁC,CŁC,PIPŁCŁ.
0014 BA 0010 R mov DX, OFFSET GREETING ; P'C<PIPsP
r PSP° CŁPeCŁP°PS PIC,PsCŁPsPN°
0017 E8 0000 R call WriteMsg ; CŁC,CŁPsP
ePŁ PiCŁPŁPIPŁC,CŁC,PIPŁCŁ.
001A CB ret ; P'C<C...PsP
r PI DOS PiPs PePsPjP°PSPPrPŁ,
; PSP°C...PsP
rCŁC%PŁPN°CŁCŁ PI 1-PsPj CŁP»PsPIPŁ PSP.
001B Main ENDP
001B CODE ENDS
END Main

```

```

__Microsoft (R) Macro Assembler Version 5.10 9/15/22
12:45:00

```

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

47990 + 459267 Bytes symbol space free

0 Warning Errors  
0 Severe Errors