

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

**ОТЧЕТ**  
**по лабораторной работе №1**  
**по дисциплине «Организация систем и ЭВМ»**  
**Тема «Трансляция, отладка и выполнение программ**  
**на языке Ассемблера»**

Студентка гр. 1303

Сырцева Д.Д

Преподаватель

Ефремов М.А

Санкт-Петербург

2022

## **Цель работы.**

Ознакомиться с тем, как происходит трансляция, отладка и выполнение программ на языке Ассемблера.

## **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. Выполняемые функцией действия и задаваемые ей параметры - следующие: - обеспечивается вывод на экран строки символов, заканчивающейся знаком "\$"; - требуется задание в регистре `ax` 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. 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, 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. Результаты

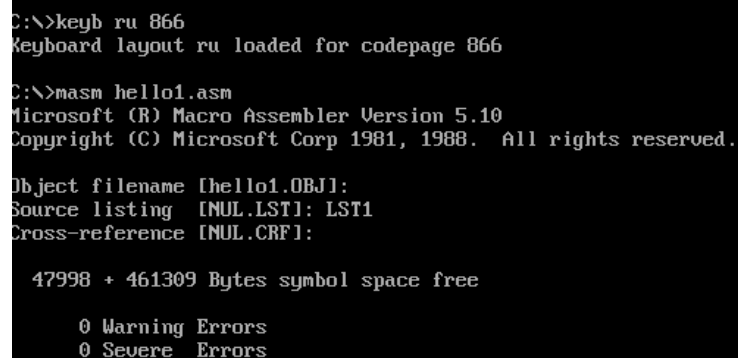
прогона программы под управлением отладчика должны быть представлены в таблице.

## 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. Протокол работы на компьютере. Протоколы пошагового исполнения каждой из программ под управлением отладчика.**

Выполнены основные пункты 1-6 части 1 настоящего задания. Программа была протранслирована «см. рис. 1» для hello1, «см. рис. 3» для hello2. Был скомпилирован загрузочный модуль с созданием карты памяти и исполняемого файла, а также была выполнена программа в автоматическом режиме «см. рис. 2» для hello1, «см. рис. 4» для hello2.



```
C:\>keyb ru 866
Keyboard layout ru loaded for codepage 866

C:\>masm hello1.asm
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.

Object filename [hello1.OBJ]:
Source listing [NUL.LST]: LST1
Cross-reference [NUL.CRF]:

47998 + 461309 Bytes symbol space free

0 Warning Errors
0 Severe Errors
```

Рисунок1

```

C:\>link hello1.obj

Microsoft (R) Overlay Linker Version 3.64
Copyright (C) Microsoft Corp 1983-1988. All rights reserved.

Run File [HELLO1.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:

C:\>hello1.exe
Вас приветствует ст.гр.1303 - Сырцева.Д.Д

```

Рисунок2

```

C:\>masm hello2.asm
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.

Object filename [hello2.OBJ]:
Source listing [NUL.LST]: LST2
Cross-reference [NUL.CRF]:

47990 + 459267 Bytes symbol space free

0 Warning Errors
0 Severe Errors

```

Рисунок3

```

C:\>link hello2.obj

Microsoft (R) Overlay Linker Version 3.64
Copyright (C) Microsoft Corp 1983-1988. All rights reserved.

Run File [HELLO2.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:

C:\>hello2.exe
Hello Worlds!
Student from 1303 - Syrtceva Daria
C:\>S

```

Рисунок4

Запущено выполнение программ под управлением отладчика.  
Полученные результаты приведены в таблицах.

Таблица 1 – Протокол пошагового исполнения hello1

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения

0000	MOV AX,1A07	B8071A	(AX) = 0000 (IP) = 0010	(AX) = 1A07 (IP) = 0013
0003	MOV DS,AX	8ED8	(DS) = 19F5 (IP) = 0013	(DS) = 1A07 (IP) = 0015
0005	MOV DX,0000	BA0000	(IP) = 0015	(IP) = 0018
0008	MOV AH,09	B409	(AX) = 1A07 (IP) = 0018	(AX) = 0907 (IP) = 001A
000A	INT 21	CD21	(IP) = 001A	(IP) = 001C
000C	MOV AH,4C	B44C	(AX) = 0907 (IP) = 001C	(AX) = 4C07 (IP) = 001E
000E	INT 21	CD21	(AX) = 4C07 (DS) = 1A07 (IP) = 001E	(AX) = 0000 (DS) = 19F5 (IP) = 0010

Таблица 2 - Протокол пошагового исполнения hello2

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0005	PUSH DS	1E	Stack(+0) = 0000 (IP) = 0005 (SP) = 0018	Stack(+0) = 19F5 (IP) = 0006 (SP) = 0016
0006	SUB AX,AX	2BC0	(IP) = 0006	(IP) = 0008
0008	PUSH AX	50	(IP) = 0008 (SP) = 0016 Stack(+0) = 19F5 Stack(+2) = 0000	(IP) = 0009 (SP) = 0014 Stack(+0) = 0000 Stack(+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	(AX) = 1A07 (IP) = 0011	(AX) = 0907 (IP) = 0014
0014	MOV DX,0010	BA1000	(IP) = 0014 (DX) = 0000	(IP) = 0017 (DX) = 0010
0017	CALL 0000	E8E6FF	(IP) = 0017	(IP) = 001A
001A	RET Far	CB	(IP) = 001A (SP) = 0014 (CS) = 1A0B Stack(+0) = 0000 Stack(+2) = 19F5	(IP) = 0000 (SP) = 0018 (CS) = 19F5 Stack(+0) = 0000 Stack(+2) = 0000
0000	INT 20	CD20	(AX) = 0907 (DX) = 0010 (CS) = 19F5 (DS) = 1A07 (IP) = 0000	(AX) = 0000 (DX) = 0000 (CS) = 1A0A (DS) = 19F5 (IP) = 0005

### **Вывод.**

Выполнена трансляция, отладка программ на языке Ассемблера.  
Разобраны структуры и реализации каждого сегмента программы.

## ПРИЛОЖЕНИЕ А

### Тексты исходных файлов программ hello1 и hello2.

```
; 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 "$";
;               - требует задания в регистре ah номера функции=09h,
;               а в регистре dx - смещения адреса выводимой
;               строки;
;               - использует регистр ax и не сохраняет его
;               содержимое.
*****
        DOSSEG                               ; Задание сегментов под ДОС
        .MODEL  SMALL                       ; Модель памяти SMALL (Малая)
        .STACK  100h                       ; Отвести под стек 256 байт
        .DATA                               ; Начало сегмента данных
Greeting LABEL BYTE                       ; Текст приветствия
        DB 'Вас приветствует ст.гр.1303 - Сырцева Д.Д.',13,10,'$'
        .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

; 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 - Syrtceva Daria$'
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

```

## Тексты файлов диагностических сообщений hello1.lst и hello2.lst.

```

; 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
0000 Greeting LABEL BYTE
; Текст приветствия
0000 C2 E0 F1 20 EF F0 DB 'Вас приветствует ст.гр.1303- Сырцева Д.Д.
; '13,10,'$'
E8 E2 E5 F2 F1 F2
E2 F3 E5 F2 20 F1
F2 2E E3 F0 2E 31
33 30 33 20 2D 20
D1 FB F0 F6 E5 E2
E0 20 C4 2E C4 2E
0D 0A 24
.CODE
; Начало сегмента кода
0000 B8---- R mov ax, @data ; Загрузка в DS адреса начала
0003 8E D8 mov ds, ax ; сегмента данных
0005 BA 0000 R mov dx, OFFSET Greeting ; Загрузка в dx смещения
; адреса текста приветствия
0008 DisplayGreeting:
0008 B4 09 mov ah, 9 ; # функции ДОС печати строки
000A CD 21 int 21h ; вывод на экран приветствия
000C B4 4C mov ah, 4ch ; # функции ДОС завершения программы
000E CD 21 int 21h ; завершение программы и выход в ДОС
END
Symbols-1

```

Segments and Groups:

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

; **HELLO2**- Учебная программа N2 лаб.раб.#1 по дисциплине "архитектура компьютера"

; Программа использует процедуру для печати строки

;

; ТЕКСТ ПРОГРАММЫ

= 0024                      EOFLine EQU '\$'        ; Определение символьной константы  
                              ;"Конец строки"

; Стек программы

                              ASSUME CS:CODE, SS:AStack  
0000                      AStack SEGMENT STACK  
0000 000C[                      DW 12 DUP('!') ; Отводится 12 слов памяти  
                              0021  
                              ]  
0018                      AStack ENDS

; Данные программы

0000                      DATA SEGMENT

; Директивы описания данных

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 1303 –Syrteceva Daria \$'  
                              74 20 66 72 6F 6D  
                              20 34 33 35 30 20  
                              2D 20 24  
0025                      DATA ENDS

; Код программы

0000                      CODE SEGMENT

; Процедура печати строки

0000                      WriteMsg PROC NEAR  
0000 B4 09                      mov AH,9  
0002 CD 21                      int 21h ; Вызов функции DOS по прерыванию  
0004 C3                      ret  
0005                      WriteMsg ENDP

; Головная процедура

0005                      Main PROC FAR  
0005 1E                      push DS ;\ Сохранение адреса начала PSP в стеке  
0006 2B C0                      sub AX,AX ; > для последующего восстановления по  
0008 50                      push AX ;/ команде ret, завершающей процедуру.

0009 B8----	R	mov AX,DATA	; Загрузка сегментного
000C 8E D8		mov DS,AX	; регистра данных.
000E BA 0000	R	mov DX, OFFSET HELLO	; Вывод на экран первой
0011 E8 0000	R	call WriteMsg	; строки приветствия.
0014 BA 0010	R	mov DX, OFFSET GREETING	; Вывод на экран второй
0017 E8 0000	R	call WriteMsg	; строки приветствия.
001A CB		ret	; Выход в DOS по команде,
			; находящейся в 1-ом слове PSP.
001B		Main	ENDP
001B		CODE	ENDS
			END Main

#### 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			