

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

**ОТЧЕТ**  
**по лабораторной работе №1**  
**по дисциплине «ОЭВМиС»**  
**Тема: Написание простейшей программы на ассемблере**

Студент гр. 0382

\_\_\_\_\_

Шангичев В. А.

Преподаватель

\_\_\_\_\_

Ефремов М. А.

Санкт-Петербург

2021

## **Цель работы.**

Разобраться в программах, написанных на низкоуровневом языке программирования ассемблер. Получить знания о компиляции и линковке таких программ.

## **Задание.**

1. Проанализировать код программы hello1.asm.

Назначение данной программы следующее:

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.

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.

### Часть 1.

- Код программы hello1 был просмотрен и тщательно разобран.
- Строка-приветствие была преобразована в соответствии с личными данными.
- Файл hello1 был загружен из каталога задания в каталог Masm.
- Программа была протранслирована с получением объектного файла и листинга. Синтаксических ошибок не обнаружено.
- Загрузочный модуль был успешно скомпонован.
- Программа была запущена в автоматическом режиме: на экран было выведено ожидаемое сообщение.
- Программа была запущена с помощью отладчика.

Ход выполнения программы представлен в таблице 1:

Начальные значения: CS = 1A05, DS = 19F5, ES = 19F5, SS = 1A0C

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До	После
0010	MOV AX, 1A07	B8 07 1A	AX = 0000 IP = 0010	AX = 1A07 IP = 0013
0013	MOV DS, AX	8E D8	DS = 19F5 IP = 0013	DS = 1A07 IP = 0015
0015	MOV DX, 0000	BA 00 00	DX = 0000 IP = 0015	DX = 0000 IP = 0018
0018	MOV AH,09	B4 09	AX = 1A07 IP = 0018	AX = 0907 IP = 001A
001A	INT 21	CD 21	IP = 001A	IP = 001C
001C	MOV AH,4C	B4 4C	AX = 0907 IP = 001C	AX = 4C07 IP = 001E
001E	INT 21	CD 21	IP = 001E	IP = 0010

Часть 2.

Для файла hello2.asm были проделаны те же шаги что и для hello1.asm. При запуске программы вывелись ожидаемые строки с приветствием. Ход выполнения программы представлен в таблице 2:

Начальные значения: CS = 1A0B, DS = 19F5, ES = 19F5, SS = 1A05

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До	После
0005	PUSH DS	1E	IP = 0005 SP = 0018	IP = 0006 SP = 0016

			Stack: + 0 0000	Stack: +0 19F5
0006	SUB AX, AX	2B C0	AX = 0000 IP = 0006	AX = 0000 IP = 0008
0008	PUSH AX	50	AX = 0000 SP = 0016 IP = 0008 Stack: +0 19F5	AX = 0000 SP = 0014 IP = 0009 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	DX = 0000 IP = 000E	DX = 0000 IP = 0011
0011	CALL 0000	E8ECFF	SP = 0014 IP = 0011 Stack: +0 0000 Stack: +2 19F5	SP = 0012 IP = 0000 Stack: +0 0014 Stack: +2 0000 Stack: +4 19F5
0000	MOV AH, 09	B4 09	AX = 1A07 IP = 0000	AX = 0907 IP = 0002
0002	INT 21	CD 21	IP = 0002	IP = 0004
0004	RET	C3	IP = 0004 SP = 0012 Stack: +0 0014 Stack: +2 0000 Stack: +4 19F5	IP = 0014 SP = 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	E8E6FF	SP = 0014 IP = 0017 Stack: +0 0000 Stack: +2 19F5	SP = 0012 IP = 0000 Stack: +0 001A Stack: +2 0000 Stack: +4 19F5
0000	MOV AH, 09	B4 09	AX = 0907 IP = 0000	AX = 0907 IP = 0002
0002	INT 21	CD 21	IP = 0002	IP = 0004
0004	RET	C3	IP = 0004 SP = 0012 Stack: +0 001A Stack: +2 0000 Stack: +4 19F5	IP = 001A SP = 0014 Stack: +0 0000 Stack: +2 19F5 Stack: +4 0000
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	IP = 0000	IP = 0005

### **Выводы.**

Были изучены и проработаны ходы двух программ, написанных на ассемблере. Были получены навыки ассемблирования и компоновки программы.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

#### Файл 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 "$";
;               - требует задания в регистре ah номера функции=09h,
;               а в регистре dx - смещения адреса выводимой
;               строки;
;               - использует регистр ax и не сохраняет его
;               содержимое.
; *****
DOSSEG                      ; Задание сегментов под ДОС
.MODEL  SMALL               ; Модель памяти-SMALL (Малая)
.STACK  100h                ; Отвести под Стек 256 байт
.DATA                      ; Начало сегмента данных
Greeting LABEL BYTE        ; Текст приветствия
DB 'Greets you st. g.0382 - Shangichev V. A.',13,10,'$'
.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
```

#### Файл 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 0382 - Shangichev Vasiliiy$'
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

```

## ПЛОЖЕНИЕ Б

### ФАЙЛЫ ДИАГНОСТИЧЕСКИХ СООБЩЕНИЙ

hello1.lst

Microsoft (R) Macro Assembler Version 5.10  
02:49:40

9/14/21

Page

1-1

```

                                DOSSEG
                                .MODEL  SMALL
                                .STACK  100h
                                .DATA
0000                Greeting  LABEL  BYTE
0000  47 72 65 65 74 73      DB 'Greets you st. g.0382 - Shangichev V.
A.
                                ',13,10,'$'
                                20 79 6F 75 20 73
                                74 2E 20 67 2E 30
                                33 38 32 20 2D 20
                                53 68 61 6E 67 69
                                63 68 65 76 20 56
                                2E 20 41 2E 0D 0A
                                24

```

Microsoft (R) Macro Assembler Version 5.10  
02:49:40

9/14/21

Page

1-2

```

                                .CODE

0000  B8 ---- R      mov  ax, @data
0003  8E D8          mov  ds, ax
0005  BA 0000 R      mov  dx, OFFSET Greeting
0008                      DisplayGreeting:
0008  B4 09          mov  ah, 9
000A  CD 21          int  21h
000C  B4 4C          mov  ah, 4ch
000E  CD 21          int  21h

```

# 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

47984 + 459276 Bytes symbol space free

0 Warning Errors  
0 Severe Errors

hello2.lst

```

= 0024                                EOFLine EQU '$'

                                ASSUME CS:CODE, SS:AStack

0000                                AStack    SEGMENT  STACK
0000 000C[                            DW 12 DUP('!')

                                0021

                                ]

0018                                AStack    ENDS

0000                                DATA      SEGMENT

                                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 0382 - Shangichev
Vasiliy$'
                                74 20 66 72 6F 6D
                                20 30 33 38 32 20
                                2D 20 53 68 61 6E
                                67 69 63 68 65 76
                                20 56 61 73 69 6C
                                69 79 24
                                0037                                DATA      ENDS

                                0000                                CODE      SEGMENT
                                0000 WriteMsg  PROC  NEAR
                                0000 B4 09                                mov    AH,9
                                0002 CD 21                                int    21h

```

, DOS

Microsoft (R) Macro Assembler Version 5.10  
02:50:28

9/14/21

1-2

```

0004  C3                                ret
0005                                WriteMsg ENDP

0005                                Main      PROC  FAR
0005  1E                                push  DS
0006  2B C0                            sub    AX,AX
0008  50                                push  AX
0009  B8 ---- R                        mov    AX,DATA
000C  8E D8                            mov    DS,AX
000E  BA 0000 R                        mov    DX, OFFSET HELLO
0011  E8 0000 R                        call   WriteMsg
0014  BA 0010 R                        mov    DX, OFFSET GREETING
0017  E8 0000 R                        call   WriteMsg
001A  CB                                ret

001B                                Main      ENDP
001B                                CODE     ENDS
                                END Main

```

Microsoft (R) Macro Assembler Version 5.10

9/14/21 02:50:28

Symbols-1

## Segments and Groups:

N a m e	Length	Align	Combine Class
ASTACK . . . . .	0018	PARA	STACK
CODE . . . . .	001B	PARA	NONE
DATA . . . . .	0037	PARA	NONE

## Symbols:

N a m e	Type	Value	Attr
EOFLINE . . . . .	NUMBER		0024
GREETING . . . . .	L BYTE		0010 DATA

	HELLO . . . . .	L BYTE	0000	DATA	
0016	MAIN . . . . .	F PROC	0005	CODE Length	=
0005	WRITEMSG . . . . .	N PROC	0000	CODE Length	=
	@CPU . . . . .	TEXT	0101h		
	@FILENAME . . . . .	TEXT	hello2		
	@VERSION . . . . .	TEXT	510		

52 Source Lines  
52 Total Lines  
13 Symbols

47976 + 459281 Bytes symbol space free

0 Warning Errors  
0 Severe Errors