

**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: ТРАНСЛЯЦИИ, ОТЛАДКА И ВЫПОЛНЕНИЕ ПРОГРАММЫ НА ЯЗЫКЕ**  
**АССЕМБЛЕРА**

Студент гр. 9383

\_\_\_\_\_

Гордон Д.А.

Преподаватель

\_\_\_\_\_

Ефремов М.А.

Санкт-Петербург

2020

### **Цель работы.**

Применить на практике знания о работе с регистрами процессора и познакомиться с основами программирования на языке ассемблер в операционной системе DOS.

### **Текст задания.**

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.

Уточнение задания следует посмотреть в файле lr1\_comp.txt каталога Задания.

#### **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 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

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.

# ПРОТОКОЛ

## Часть 1.

Hello1.asm
Программа просмотрена.
Разобрался в структуре программы, данные строки-приветствия были изменены.
Файл загружен.
Ошибки обнаружены не были.
Загрузочный модуль скомпонован, карта памяти записана в файл hello1.map.
Программа завершилась корректно, на экран было выведено сообщение: «Вас приветствует ст.гр.7303 - Иванов И.И.»

Hello2.asm
Программа просмотрена.
Разобрался в структуре программы, данные строки-приветствия были изменены.
Файл загружен.
Ошибки были в строчке 28 — отсутствовали запятые при многократном вызове директивы ASSUME.
Загрузочный модуль скомпонован, карта памяти записана в файл hello2.map.
Программа завершилась корректно, на экран было выведено сообщение: «Hello Worlds! Student from 4350 - ».

## Часть 2.

### hello1.exe

**CS = 1A05, DS = 19F5, ES = 19F5, SS = 1A0A, SP = 0100**

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0010	MOV AX, 1A07	B8071A	AX = 0000 IP = 0010	AX = 1A07 IP = 0013
0013	MOV DS, AX	8ED8	AX = 1A07	AX = 1A07

			DS = 19F5 IP = 0013	DS = 1A07 IP = 0015
0015	MOV DX, 0000	BA0000	DX = 0000 IP = 0015	DX = 0000 IP = 0018
0018	MOV AH, 09	B409	AH = 1A IP = 0018	AH = 09 IP = 001A
001A	INT 21	CD21	IP = 001A	IP = 001C
001C	MOV AH, 4C	B44C	AH = 09 IP = 001C	AH = 4C IP = 001E
001E	INT 21	CD21	IP = 001E	

### hello2.exe

CS = 1A0A, DS = 19F5, ES = 19F5, SS = 1A05, SP = 0018

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0005	PUSH DS	1E	DS = 19F5 SP = 0018	DS = 19F5 SP = 0016
0006	SUB AX, AX	2BC0	AX = 0000	AX = 0000
0008	PUSH AX	50	AX = 0000 SP = 0016	AX = 0000 SP = 0014
0009	MOV AX, 1A07	B8071A	AX = 0000	AX = 1A07
000C	MOV DS, AX	8ED8	DS = 19F5	DS = 1A07
000E	MOV DX, 0000	BA0000	DX = 0000	DX = 0000
0011	CALL 0000	E8ECFF	AX = 1A07 SP = 0014	AX = 0907 SP = 0012
0000	MOV AH, 09	B409	AH = 1A	AH = 09
0002	INT 21	CD21		
0004	RET	C3	SP = 0012	SP = 0014
0014	MOV DX, 0010	BA1000	DX = 0000	DX = 0010

0017	CALL 0000	E8E6FF	SP = 0014	SP = 0012
0000	MOV AH, 09	B409	AH = 09	AH = 09
0002	INT 21	CD21		
0004	RET	C3	SP = 0012	SP = 0014
001A	RET Far	CB	CS = 1A0A	CS = 19F5
0000	INT 20	CD20		

PUSH – занести значение регистров и ячеек памяти в стек по адресу SS:SP

MOV A, B – переместить значение из B → A

RET – возврат в программу

RET FAR – процедуры FAR вы можете вызывать

вне сегмента, в котором они определяются. Вызов FAR заносит в

стек адрес в виде сегмента и смещения, а затем устанавливает

CS:IP в адрес процедуры. Когда процессор обнаруживает возврат

дальнего типа, он извлекает из стека сегмент и смещение адреса возврата и устанавливает в него CS:IP .

RET NEAR – возвращает в тот сегмент, где была определена процедура. Вызов ближнего типа заносит адрес возврата в стек и устанавливает IP в значение смещения процедуры.

SUB A, B – из A вычесть B и записать в A.

CALL x – передаёт управление команде, находящейся по адресу x.

DB – байт.

PSP – структура данных, где хранится состояние программы. Адрес хранится в DS.

INT 21h – функция DOS -- вызов прерывания (считывает из AH номер прерывания).

OFFSET – адрес переменной

## **ВЫВОДЫ**

Познакомился с основами программирования на языке Assembly.

## ПРИЛОЖЕНИЕ

*Hello1.asm:*

```

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.7303 - Иванов И.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

```

*Hello1.lst:*

```

                                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 82 A0 E1 20 AF E0 DB 'Вас приветствует ст.гр.7303 - Иванов И.И
                                .',13,10','$'

                                A8 A2 A5 E2 E1 E2
                                A2 E3 A5 E2 20 E1
                                E2 2E A3 E0 2E 37
                                33 30 33 20 2D 20
                                88 A2 A0 AD AE A2
                                20 88 2E 88 2E 0D
                                0A 24

                                .CODE                      ; Начал
                                о сегмента кода

```



0000 B8 ---- R	mov ax, @data	; Загр
	узка в DS адреса начала	
0003 8E D8	mov ds, ax	; сегм
	ента данных	
0005 BA 0000 R	mov dx, OFFSET Greeting	; Загр
	узка в dx смещения	

```
                                ; адрес
                                а текста приветствия
0008 DisplayGreeting:
0008 B4 09      mov ah, 9          ; # фу
                                нкции ДОС печати строки
000A CD 21      int 21h          ; вывод
                                на экран приветствия
000C B4 4C      mov ah, 4ch      ; # фу
                                нкции ДОС завершения программы
000E CD 21      int 21h          ; завер
                                шение программы и выход в ДОС
                                END
```

Segments and Groups:

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

48006 + 461301 Bytes symbol space free

0 Warning Errors  
0 Severe Errors

*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

AStack SEGMENT STACK

DW 12 DUP(?) ; Отводится 12 слов памяти (массив из 12  
неинициализированных по 2 байт, т.е. 24 байт)

AStack ENDS

; Данные программы

DATA SEGMENT

; Директивы описания данных

HELLO DB 'Hello Worlds!', 0AH, 0DH,EOFLine ;массив?

GREETING DB 'Student from 4350 - \$'

DATA ENDS

; Код программы

CODE SEGMENT

ASSUME CS:Code DS:DATA SS:AStack

; 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

```

*Hello2.lst:*

```

; HELLO2 - Учебная программа N2 лаб.раб.#1 по
; дисциплине "Архитектура компьютера"
; Программа использует процедуру для п
; ечати строки
;
; ТЕКСТ ПРОГРАММЫ

= 0024          EOFLine EQU '$'      ; Определение символъ
              ной константы
              ; "Конец строки"

; стек программы

0000          AStack  SEGMENT STACK
0000 000C[      DW 12 DUP(?) ; Отводится 12 слов п
              амяти (массив из 12 неинициализированных по 2 б
              айт, т.е. 24 байт)
              ???
              ]

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
              ;массив?
              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

; Код программы

```

```

0000          CODE    SEGMENT
                ASSUME CS:Code DS:DATA SS:AStack
hello2.asm(28): warning A4001: Extra characters on line
                ; Процедура печати строки
0000          WriteMsg PROC NEAR
0000 B4 09                mov  AH,9
0002 CD 21                int  21h ; Вызов функции DOS по пре
                рыванию
0004 C3                ret
0005          WriteMsg ENDP

                ; Главная процедура
0005          Main    PROC FAR
0005 1E                push DS    ;\ Сохранение адреса
                начала PSP в стеке

```

```
0006 2B C0                sub  AX,AX    ; > для последующего в
                                остановления по
0008 50                    push AX    ;/ команде ret, завер
                                шающей 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 пр
                                иветствия.
0014 BA 0010 R            mov  DX, OFFSET GREETING ; Вывод на
                                экран второй
0017 E8 0000 R            call WriteMsg    ; строки пр
                                иветствия.
001A CB                    ret            ; Выход в D
                                OS по команде,
                                ; находящей
                                ся 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		

51 Source Lines  
51 Total Lines  
13 Symbols

48014 + 461293 Bytes symbol space free

1 Warning Errors  
0 Severe Errors