

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

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

Студент гр. 0382

\_\_\_\_\_

Тихонов С.В.

Преподаватель

\_\_\_\_\_

Ефремов М. А.

Санкт-Петербург

2021

### **Цель работы.**

Изучить трансляцию, отладку и выполнение программ на языке Ассемблера

### **Задание.**

#### **Вариант 2**

##### **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 4 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

- 1) Изучена программа hello1.asm. Строка приветствия была изменена в соответствии с личными данными.
- 2) Выполнена трансляция hello1.asm в hello1.obj с созданием файла листинга. Трансляция прошла без ошибок.
- 3) Выполнена компоновка объектного файла с созданием карты памяти и исполняемого файла hello.exe
- 4) Выполнен запуск исполняемого файла в автоматическом режиме. Результатом работы является строка: "hello everybody, its Tikhonov Sergey from 0382"
- 5) Выполнен запуск исполняемого файла с помощью отладчика.

Таблица 1. Результаты выполнения hello1.exe в отладчике

Начальные значения сегментных регистров: (CS) = 1A05, (DS) = 19F5, (ES) = 19F5, (SS) = 1A0A

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	(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	(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	(IP) = 001E	(IP) = 0010

## Часть 2.

Проделаем аналогичные шаги для программы hello2.exe

При запуске на экран было выведено: “Hello Worlds! \n Student from 0382 — Tikhonov Sergey”

Начальное значение 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 (SP) = 0018 Stack +0 0000 +2 0000 +4 0000 +6 0000	(IP) = 0006 (SP) = 0016 Stack +0 19 F5 +2 0000 +4 0000 +6 0000
0006	SUB AX, AX	2BC0	(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	BED8	(DS) = 19F5 (AX) = 1A07 (IP) = 000C	(DS) = 1A07 (AX) = 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 +2 0000 +4 19F5
0000	MOV AH, 9	B409	(AX) = 1A07 (IP) = 0000	(AX) = 0907 (IP) = 0002
0002	INT 21	CD21	(IP) = 0002	(IP) = 0004
0004	RET	C3	(IP) = 0004	(IP) = 0014

			(SP) = 0012 Stack: +0 0014 +2 0000 +4 19 F5 +60000	(SP) = 0014 Stack: +0 0000 +2 19F5 +40000 +60000
0014	MOV DX, 0010	BA1000	(DX) = 0000 (IP) = 0014	(DX) = 0010 (IP) = 0017
0017	CALL 0000	E6FF	(SP) = 0014 (IP) = 0017 Stack: +0 0000 +2 19F5 +40000 +60000	(SP) = 0012 (IP) = 0000 Stack: +0 001A +2 0000 +4 19F5 +60000
0000	MOV AH, 9	B409	(AX) = 1A07 (IP) = 0000	(AX) = 0907 (IP) = 0002
0002	INT 21	CD21	(IP) = 0002	(IP) = 0004
0004	RET	C3	(IP) = 0004 (SP) = 0012 Stack: +0 001A +2 0000 +4 19F5	(IP) = 001A (SP) = 0014 Stack: +0 0000 +2 19F5
001A	RET Far	CB	(IP) = 001A (SP) = 0014 (CS) = 1A0B	(IP) = 0000 (SP) = 0018 (CS) = 19F5
0000	INT 20	CD 20	(IP) = 0000	(IP) = 0005

### **Выводы.**

Были изучены основные элементы синтаксиса ассемблера и правила написания программ на нём. Та же были рассмотрены процессы трансляции, компоновки и выполнения программы.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

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 'hello everybody, its Tikhonov Sergey from 0382',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 0382 - Tikhonov Sergey $'
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