

**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 Ассембле-**  
**ра.**

Студент гр. 9383

\_\_\_\_\_

Ноздрин В.Я.

Преподаватель

\_\_\_\_\_

Ефремов М.А.

Санкт-Петербург

2020

## **Цель работы.**

Изучить основные принципы трансляции, отладки и выполнения программ на языке Ассемблера.

## **Ход работы.**

### **Часть 1.**

1. Загружены файлы из каталога /tools в каталог ~/tools
2. Запущена программа DOSBox, смонтирован виртуальный диск C: в каталоге ~/tools при помощи mount c ~/tools
3. Просмотрен код программ hello1.asm и hello2.asm, изучены структура и реализация сегментов программы. Строка-приветствие преобразована в соответствии с личными данными.
4. В DOS осуществлен переход на виртуальный диск при помощи команды c:
5. Протранслирована программа с помощью строки:  
> nasm hello1.asm  
Создается объектный файл Hello1.obj с сообщением об отсутствии ошибок и предупреждений.
6. Скомпонован загрузочный модуль Hello1.exe с помощью строки:  
> link Hello1.obj  
Линковщик создает загрузочный модуль Hello1.exe.
7. Загружена русская кодовая таблица символов путём набора строки:  
> keyb ru 866
8. Запущена программа в автоматическом режиме путем набора строки:  
> Hello1.exe
9. Вывод программы:  
> Вас приветствует ст.гр.9383 — Ноздрин В. Я.

10. Выполнен запуск программы Hello1.exe в пошаговом режиме с фиксацией используемых регистров и ячеек памяти до и после выполнения каждой команды, используя отладчик и соответственно команду:

> afdpro Hello1.exe.

Продвижение по сегментам экранной формы отладчика выполняется с помощью клавиш F7 – F10 (up, down, left, right). Перезапуск программы в отладчике выполняется клавишей F3 (Retrieve). Выход из отладчика - по команде Quit.

1. Содержимое сегментных регистров до старта программы:

CS:1A05, DS:19F5, ES:19F5, SS:1A0A, HS:19F5, FS:19F5

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

**Часть 2.**

1. Просмотрен код программы hello2.asm, изучены структура и реализация сегментов программы. Строки-приветствия преобразованы в соответствии с личными данными.

2. Выполнена трансляция программы hello2.asm командой:

>masm hello2.asm

Создан объектный файл hello2.obj с сообщением об отсутствии ошибок.

3. Скомпилирован загрузочный модуль hello2.exe командой:

>link Hello2.obj

4. Выполнена программа Hello2.exe в автоматическом режиме и проконтролировано, что она работает корректно: в консоль выводится:

>Hello Worlds!

>Student from 9383 - Nozdrin V.Y..

5. Запущена программа Hello2.exe в пошаговом режиме, используя отладчик afddpro с фиксацией содержимого используемых регистров и ячеек памяти до и после выполнения команд.

Содержимое сегментных регистров до старта программы: CS:1A0A, DS:19F5, ES:19F5, SS:1A05, HS:19F5, FS: 19F5.

Табл.2.

Адрес Команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0005	PUSH DS	1E	(AX) = 0000 (DS) = 19F5 (SP) = 0018 (IP) = 0005 Stack +0 0000 +2 0000 +4 0000	(AX) = 0000 (DS) = 19F5 (SP) = 0016 (IP) = 0006 Stack +0 19F5 +2 0000 +4 0000
0006	SUB AX, AX	2BC0	(AX) = 0000 (DS) = 19F5 (SP) = 0016 (IP) = 0006 Stack +0 19F5 +2 0000 +4 0000	(AX) = 0000 (DS) = 19F5 (SP) = 0016 (IP) = 0008 Stack +0 19F5 +2 0000 +4 0000

0008	PUSH AX	50	(AX) = 0000 (DS) = 19F5 (SP) = 0016 (IP) = 0008 Stack +0 19F5 +2 0000 +4 0000	(AX) = 0000 (DS) = 19F5 (SP) = 0014 (IP) = 0009 Stack +0 0000 +2 19F5 +4 0000
0009	MOV AX, 1A07	B8071A	(AX) = 0000 (DS) = 19F5 (SP) = 0014 (IP) = 0009 Stack +0 0000 +2 19F5 +4 0000	(AX) = 1A07 (DS) = 19F5 (SP) = 0014 (IP) = 000C Stack +0 0000 +2 19F5 +4 0000
000C	MOV DS, AX	8ED8	(AX) = 1A07 (DS) = 19F5 (SP) = 0014 (IP) = 000C Stack +0 0000 +2 19F5 +4 0000	(AX) = 1A07 (DS) = 1A07 (SP) = 0014 (IP) = 000E Stack +0 0000 +2 19F5 +4 0000
000E	MOV DX, 0000	BA0000	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (SP) = 0014 (IP) = 000E Stack +0 0000 +2 19F5 +4 0000	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (SP) = 0014 (IP) = 0011 Stack +0 0000 +2 19F5 +4 0000
0011	CALL 0000	E8ECFF	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (SP) = 0014 (IP) = 0011 Stack +0 0000 +2 19F5 +4 0000	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (SP) = 0012 (IP) = 0000 Stack +0 0014 +2 0000 +4 19F5
0000	MOV AH,09	B409	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (SP) = 0012 (IP) = 0000 Stack +0 0014 +2 0000 +4 19F5	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (SP) = 0012 (IP) = 0002 Stack +0 0014 +2 0000 +4 19F5

0002	INT 21	CD21	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (SP) = 0012 (IP) = 0002 Stack +0 0014 +2 0000 +4 19F5	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (SP) = 0012 (IP) = 0004 Stack +0 0014 +2 0000 +4 19F5
0004	RET	C3	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (SP) = 0012 (IP) = 0004 Stack +0 0014 +2 0000 +4 19F5	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (SP) = 0014 (IP) = 0014 Stack +0 0000 +2 19F5 +4 0000
0014	MOV DX,0010	BA1000	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (SP) = 0014 (IP) = 0014 Stack +0 0000 +2 19F5 +4 0000	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (SP) = 0014 (IP) = 0017 Stack +0 0000 +2 19F5 +4 0000
0017	CALL 0000	E8E6FF	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (SP) = 0014 (IP) = 0017 Stack +0 0000 +2 19F5 +4 0000	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (SP) = 0012 (IP) = 0000 Stack +0 001A +2 0000 +4 19F5
0000	MOV AH,09	B409	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (SP) = 0012 (IP) = 0000 Stack +0 001A +2 0000 +4 19F5	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (SP) = 0012 (IP) = 0002 Stack +0 001A +2 0000 +4 19F5
0002	INT 21	CD21	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (SP) = 0012 (IP) = 0002 Stack +0 001A +2 0000 +4 19F5	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (SP) = 0012 (IP) = 0004 Stack +0 001A +2 0000 +4 19F5

0004	RET	C3	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (SP) = 0012 (IP) = 0004 Stack +0 001A +2 0000 +4 19F5	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (SP) = 0014 (IP) = 001A Stack +0 0000 +2 19F5 +4 0000
001A	RET Far	CB	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (SP) = 0014 (IP) = 001A Stack +0 0000 +2 19F5 +4 0000	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (SP) = 0018 (IP) = 0000 Stack +0 0000 +2 0000 +4 0000
0000	INT 20	CD20	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (SP) = 0018 (IP) = 0000 Stack +0 0000 +2 0000 +4 0000	(AX) = 0000 (DX) = 0000 (DS) = 19F5 (SP) = 0018 (IP) = 0005 Stack +0 0000 +2 0000 +4 0000

### **Выводы.**

В результате выполнения лабораторной работы была освоена трансляция, изучено, как происходит выполнение и отладка программ на языке Ассемблера, а также разобраны структуры приведенных в работе программ.

## Приложение.

### Текст файла *hello1.asm*

```
N1      ; HELLO1.ASM - упрощенная версия учебной программы лаб.раб.
        ;
        ; по дисциплине "Архитектура компьютера"
        ;
        *****
*
        ; Назначение: Программа формирует и выводит на экран привет-
ствие
        ;
        ; пользователя с помощью функции ДОС "Вывод
строки"
        ;
        ; (номер 09 прерывание 21h), которая:
        ;
        ; - обеспечивает вывод на экран строки симво-
лов,
        ;
        ; заканчивающейся знаком "$";
        ;
        ; - требует задания в регистре ah номера функ-
ции=09h,
        ;
        ; а в регистре dx - смещения адреса выво-
димой
        ;
        ; 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.9383 - Ноздрин В.Я.',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
```



### ***Текст 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

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 9383 - Nozdrin V.Y.\$'  
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

```

дурю.      push  AX          ;/  команде ret, завершающей проце-
           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

```