

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

**ОТЧЁТ**  
**по лабораторной работе №1**  
**по дисциплине «Организация ЭВМ и систем»**  
**ТЕМА: ТРАНСЛЯЦИИ, ОТЛАДКА И ВЫПОЛНЕНИЕ ПРОГРАММ НА ЯЗЫКЕ**  
**ASSEMBLER.**

Студент гр. 1303

Кузнецов Н.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

### **Цель работы.**

Изучить основные принципы трансляции, отладки и выполнения программ на языке ассемблера процессора Intel x86.

### **Задание.**

1. Просмотреть программу hello1.asm.
2. Разобраться в структуре и реализации каждого сегмента программы.

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 (файла листинга). 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.

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

1. Просмотрена программа hello1.asm.
2. Разобрана структура и реализация каждого сегмента программы.

Строка-приветствие преобразована в соответствии с личными данными.

3. Загружен файл hello1.asm из каталога Задания в каталог Masm.

4. Протранслирована программа с помощью команды

> masm hello1.asm

с созданием объектного файла (hello1.obj) и файла диагностических сообщений (файла листинга).

5. Скомпонован загрузочный модуль с помощью строки

> link hello1.obj

с созданием карты памяти и исполняемого файла hello1.exe.

6. Выполнена программа в автоматическом режиме, результат

корректен

```
C:\>HELLO1.EXE
Student Kuznetsov N.A. from 1303 greeting u!
```

7. Запущено выполнение программы под управлением отладчика с

помощью команды

> afd hello1.exe

Последовательность команд и состояний hello1.exe представлена в таблице 1.

Таблица 1:

No	Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
				До	После
1.	0010	MOV AX,1A07	B8071A	AX = 0000 IP = 0010	AX = 1A07 IP = 0013
2.	0013	MOV DS,AX	8ED8	DS = 19F5 IP = 0013	DS = 1A07 IP = 0015

3.	0015	MOV DX,0000	BA0000	IP = 0015 DX = 0000	IP = 0018 DX = 0000
4.	0018	MOV AH,09	B409	AX = 1A07 IP = 0018	AX = 0907 IP = 001A
5.	0001A	INT 21	CD21	IP = 001A	IP = 001C
6.	0001C	MOV AH,4C	B44C	AX = 0907 IP = 001C	AX = 4C07 IP = 001E
7.	0001E	INT 21	CD21	AX = 4C07 IP = 001E CX = 0052 DS = 1A07	AX = 0000 IP = 0010 CX = 0000 DS = 19F5

1. Просмотрена программа hello2.asm.

2. Разобрана структура и реализация каждого сегмента программы.

Строка-приветствие преобразована в соответствии с личными данными.

3. Загружен файл hello2.asm из каталога Задания в каталог Masm.

4. Протранслирована программа с помощью команды

> masm hello1.asm

с созданием объектного файла (hello2.obj) и файла диагностических сообщений (файла листинга).

5. Скомпонован загрузочный модуль с помощью строки

> link hello2.obj

с созданием карты памяти и исполняемого файла hello2.exe.

6. Выполнена программа в автоматическом режиме, результат корректен

```
C:\>HELLO2.EXE
Hello Worlds!
Student from 1303 - Kuznetsov Nick
```

7. Запущено выполнение программы под управлением отладчика с помощью команды

> afd hello2.exe

Последовательность команд и состояний hello2.exe представлена в таблице 2.

Таблица 2:

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До	После
0005	PUSH DS	1E	SP = 0018 IP = 0005 Stack +0 0000	SP = 0016 IP = 0006 Stack +0 19F5
0006	SUB AX,AX	2BC0	IP = 0006	IP = 0008
0008	PUSH AX	50	IP = 0008 SP = 0016 Stack +0 19F5 +2 0000	IP = 0009 SP = 0014 Stack +0 0000 +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 DX = 0000	IP = 0011 DX = 0000
0011	CALL 0000	E8ECFF	SP = 0014 IP = 0011 Stack +0 0000 +2 19F5 +4 0000	SP = 0012 IP = 0000 Stack +0 0014 +2 0000 +4 19F5
0000	MOV AH,09	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 19F5	SP = 0014 Stack +0 0000 +2 19F5 +4 0000
0014	MOV DX,0010	BA1000	DX = 0000 IP = 0014	DX = 0010 IP = 0017
0017	CALL 0000	E8E6FF	IP = 0017 SP = 0014 Stack +0 0000 +2 19F5 +4 0000	IP = 0000 SP = 0012 Stack +0 001A +2 0000 +4 19F5
0000	MOV AH,09	B409	AX = 0907 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 = 0014 SP = 0014 Stack +0 0000 +2 19F5 +4 0000
001A	RET far	CB	SP = 0014 IP = 001A CS = 1A0B Stack +0 0000 +2 19F5 +4 0000	SP = 0018 IP = 0000 CS = 19F5 Stack +0 0000 +2 0000 +4 0000
0000	INT 20	CD 20	AX = 0907 CX = 007B DX = 0010 CS = 19F5 DS = 1A07 IP = 0000	AX = 0000 CX = 0000 DX = 0000 CS = 1A0B DS = 19F5 IP = 0005

## **Выводы**

В ходе данной работы была разобрана структура и реализация каждого сегмента программ. Были изучены процессы трансляции, отладки и выполнения программ на языке Ассемблера.

## 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 'Student Kuznetsov N.A. from 1303 greeting u!',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 1303 - Kuznetsov Nick$'
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

; Главная 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

```