

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

**Кафедра Математического обеспечения электронно-вычислительных
машин**

ОТЧЕТ
по лабораторной работе №1
по дисциплине «Организация ЭВМ и систем»
ТЕМА: ТРАНСЛЯЦИЯ, ОТЛАДКА И ВЫПОЛНЕНИЕ ПРОГРАММ НА ЯЗЫКЕ
АССЕМБЛЕРА.

Студентка гр. 0382

Рубежова Н.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

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

Ход выполнения.

Часть 1.

1. Протранслируем программу с созданием объектного файла *hello1.obj* и файла листинга *hello1.lst* с помощью строки :

```
> masm hello1.asm
```

2. Скомпилируем загрузочный модуль с созданием исполняемого файла с помощью строки:

```
> link hello1.obj
```

3. Выполним программу:

```
> hello1.exe
```

4. В результате выполнения программы вывелась строка-приветствие '*You are welcomed by st. gr. 0382-Rubezhova N.A.*', что полностью подтверждает корректность работы программы *hello1.asm*.

5. Запустим выполнение программы под управлением отладчика *afopro*:

```
> afopro hello1.exe
```

6. Зафиксируем начальное содержание сегментных регистров:

(CS)=1A05, (DS)=19F5, (ES)=19F5, (SS)=1A0B

7. Выполним программу в пошаговом режиме с фиксацией используемых регистров и ячеек памяти до и после выполнения каждой команды. Результаты прогона программы представлены в табл. 1.

Табл.1.

Адрес команды	Символический код команды	16-ичный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0010	MOV AX, 1A07	B8 07 1A	(AX)=0000 (IP)=0010	(AX)=1A07 (IP)=0013
0013	MOV DS, AX	8E D8	(AX) = 1A07 (DS) = 19F5 (IP) = 0013	(AX) = 1A07 (DS) = 1A07 (IP) = 0015

0015	MOV DX, 0000	BA 00 00	(IP)=0015	(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	(AX)=4C07 (CX)=0052 (DS)=1A07 (IP)=001E	(AX)=0000 (CX)=0000 (DS)=19F5 (IP)=0010

Часть 2.

1. Протранслируем программу с созданием объектного файла *hello2.obj* и файла листинга *hello2.lst* с помощью строки :

```
> masm hello2.asm
```

2. Скомпилируем загрузочный модуль с созданием исполняемого файла с помощью строки:

```
> link hello2.obj
```

3. Выполним программу:

```
> hello2.exe
```

4. В результате выполнения программы вывелись две строки 'Hello Worlds!' и 'Student from 0382 – Rubezhova N.A.', что полностью подтверждает корректность работы программы *hello2.asm*.

5. Запустим выполнение программы под управлением отладчика *afdpro*:

```
> afdpro hello2.exe
```

6. Зафиксируем начальное содержание сегментных регистров:

```
(CS)=1A0B, (DS)=19F5, (ES)=19F5, (SS)=1A05
```

7. Выполним программу в пошаговом режиме с фиксацией используемых регистров и ячеек памяти до и после выполнения каждой команды. Результаты прогона программы представлены в табл. 2.

Табл.2

Адрес команды	Символический код команды	16-ичный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения

0005	PUSH DS	1E	(SP)=0018 (IP)=0005	(SP)=0016 (IP)=0006
0006	SUB AX, AX	2B C0	(IP) = 0006	(IP) = 0008
0008	PUSH AX	50	(SP)=0016 (IP)=0008	(SP)=0014 (IP)=0009
0009	MOV AX, 1A07	B8 07 1A	(AX)=0000 (IP)=0009	(AX)=1A07 (IP)=000C
000C	MOV DS, AX	8E D8	(DS)=19F5 (IP)=000C	(DS)=1A07 (IP)=000E
000E	MOV DX, 0000	BA 00 00	(IP)=000E	(IP)=0011
0011	CALL 0000	E8 EC FF	(SP)=0014 (IP)=0011	(SP)=0012 (IP)=0000
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	(SP)=0012 (IP) = 0004	(SP) = 0014 (IP) = 0014
0014	MOV DX, 0010	BA 10 00	(DX)=0000 (IP)=0014	(DX)=0010 (IP)=0017
0017	CALL 0000	E8 E6 FF	(SP)=0014 (IP)=0017	(SP)=0012 (IP)=0000
0000	MOV AH, 09	B4 09	(IP)=0000	(IP)=0002
0002	INT 21	CD 21	(IP)=0002	(IP)=0004
0004	RET	C3	(SP)=0012 (IP)=0004	(SP)=0014 (IP)=001A
001A	RET Far	CB	(CS)=1A0B (SP)=0014 (IP)=001A	(CS)=19F5 (SP)=0018 (IP)=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.

Были изучены, а также отработаны на практике основные принципы 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 'You are welcomed by st. gr. 0382-Rubezhova N.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
```

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 - Rubeshova N.A.$'
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

```