

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. 8381

Юрьев С.Ю.

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 hello1.asm, hello2.asm, masm.exe, link.exe, afd.com, lib.exe из каталога \лаборат_работы в каталог D:\asmlab.

2. Запущена программа DOSBox, смонтирован виртуальный диск C: в каталоге \assembler1lab при помощи mount C ~\assembler1lab .

3. Просмотрена программа в режиме редактирования, изучена структура и реализация каждого сегмента 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:

> masm hello1.asm

По ходу трансляции создается объектный файл Hello1.obj. Во время её выполнения ошибок не было обнаружено.

6. Скомпонован загрузочный модуль Hello1.exe с помощью строки:

> link Hello1.obj

В результате работы линковщика создается 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.9382 — Юрьев Сергей.

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.exe 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. Просмотрена программа Hello2.asm в режиме редактирования, изучена ее структура и реализация каждого сегмента программы. Строки-приветствия преобразованы в соответствии с личными данными.

2. Выполнена трансляция программы Hello2.asm с помощью транслятора MASM и команды:

>masm Hello2.asm

В результате чего получился объектный файл Hello2.asm. В процессе трансляции ошибок не обнаружено.

3. Используя линковщик LINK, создан загрузочный модуль Hello2.exe. Использована команда:

>link Hello2.obj

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

>Hello Worlds!

>Yuryev Sergey from 9382.

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

Содержимое сегментных регистров до старта программы: CS:1A08, 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) = 0000 (IP) = 0005 Stack +0 6548 +2 6C6C +4 206F +6 6F57	(AX) = 0000 (DS) = 19F5 (SP) = FFFE (IP) = 0006 Stack +0 19F5 +2 6548 +4 6C6C +6 206F

0006	SUB AX, AX	2BC0	(AX) = 0000 (DS) = 19F5 (SP) = FFFE (IP) = 0006 Stack +0 19F5 +2 6548 +4 6C6C +6 206F	(AX) = 0000 (DS) = 19F5 (SP) = FFFE (IP) = 0008 Stack +0 19F5 +2 6548 +4 6C6C +6 206F
0008	PUSH AX	50	(AX) = 0000 (DS) = 19F5 (SP) = FFFE (IP) = 0008 Stack +0 19F5 +2 6548 +4 6C6C +6 206F	(AX) = 0000 (DS) = 19F5 (SP) = FFFC (IP) = 0009 Stack +0 0000 +2 19F5 +4 6548 +6 6C6C
0009	MOV AX, 1A05	B8051A	(AX) = 0000 (DS) = 19F5 (SP) = FFFC (IP) = 0009 Stack +0 0000 +2 19F5 +4 6548 +6 6C6C	(AX) = 1A05 (DS) = 19F5 (SP) = FFFC (IP) = 000C Stack +0 0000 +2 19F5 +4 6548 +6 6C6C
000C	MOV DS, AX	8ED8	(AX) = 1A05 (DS) = 19F5 (SP) = FFFC (IP) = 000C Stack +0 0000 +2 19F5 +4 6548 +6 6C6C	(AX) = 1A05 (DS) = 1A05 (SP) = FFFC (IP) = 000E Stack +0 0000 +2 19F5 +4 6548 +6 6C6C
000E	MOV DX, 0000	BA0000	(AX) = 1A05 (DX) = 0000 (DS) = 1A05 (SP) = FFFC (IP) = 000E Stack +0 0000 +2 19F5 +4 6548 +6 6C6C	(AX) = 1A05 (DX) = 0000 (DS) = 1A05 (SP) = FFFC (IP) = 0011 Stack +0 0000 +2 19F5 +4 6548 +6 6C6C

0011	CALL 0000	E8ECFF	(AX) = 1A05 (DX) = 0000 (DS) = 1A05 (SP) = FFFC (IP) = 0011 Stack +0 0000 +2 19F5 +4 6548 +6 6C6C	(AX) = 1A05 (DX) = 0000 (DS) = 1A05 (SP) = FFFA (IP) = 0000 Stack +0 0014 +2 0000 +4 19F5 +6 6548
0000	MOV AH,09	B409	(AX) = 1A05 (DX) = 0000 (DS) = 1A05 (SP) = FFFA (IP) = 0000 Stack +0 0014 +2 0000 +4 19F5 +6 6548	(AX) = 0905 (DX) = 0000 (DS) = 1A05 (SP) = FFFA (IP) = 0002 Stack +0 0014 +2 0000 +4 19F5 +6 6548
0002	INT 21	CD21	(AX) = 0905 (DX) = 0000 (DS) = 1A05 (SP) = FFFA (IP) = 0002 Stack +0 0014 +2 0000 +4 19F5 +6 6548	(AX) = 0905 (DX) = 0000 (DS) = 1A05 (SP) = FFFA (IP) = 0004 Stack +0 0014 +2 0000 +4 19F5 +6 6548
0004	RET	C3	(AX) = 0905 (DX) = 0000 (DS) = 1A05 (SP) = FFFA (IP) = 0004 Stack +0 0014 +2 0000 +4 19F5 +6 6548	(AX) = 0905 (DX) = 0000 (DS) = 1A05 (SP) = FFFC (IP) = 0014 Stack +0 0000 +2 19F5 +4 6548 +6 6C6C
0014	MOV DX,0010	BA1000	(AX) = 0905 (DX) = 0000 (DS) = 1A05 (SP) = FFFC (IP) = 0014 Stack +0 0000 +2 19F5 +4 6548 +6 6C6C	(AX) = 0905 (DX) = 0010 (DS) = 1A05 (SP) = FFFC (IP) = 0017 Stack +0 0000 +2 19F5 +4 6548 +6 6C6C

0017	CALL 0000	E8E6FF	(AX) = 0905 (DX) = 0010 (DS) = 1A05 (SP) = FFFC (IP) = 0017 Stack +0 0000 +2 19F5 +4 6548 +6 6C6C	(AX) = 0905 (DX) = 0010 (DS) = 1A05 (SP) = FFFA (IP) = 0000 Stack +0 001A +2 0000 +4 19F5 +6 6548
0000	MOV AH,09	B409	(AX) = 0905 (DX) = 0010 (DS) = 1A05 (SP) = FFFA (IP) = 0000 Stack +0 001A +2 0000 +4 19F5 +6 6548	(AX) = 0905 (DX) = 0010 (DS) = 1A05 (SP) = FFFA (IP) = 0002 Stack +0 001A +2 0000 +4 19F5 +6 6548
0002	INT 21	CD21	(AX) = 0905 (DX) = 0010 (DS) = 1A05 (SP) = FFFA (IP) = 0002 Stack +0 001A +2 0000 +4 19F5 +6 6548	(AX) = 0905 (DX) = 0010 (DS) = 1A05 (SP) = FFFA (IP) = 0004 Stack +0 001A +2 0000 +4 19F5 +6 6548
0004	RET	C3	(AX) = 0905 (DX) = 0010 (DS) = 1A05 (SP) = FFFA (IP) = 0004 Stack +0 001A +2 0000 +4 19F5 +6 6548	(AX) = 0905 (DX) = 0010 (DS) = 1A05 (SP) = FFFC (IP) = 001A Stack +0 0000 +2 19F5 +4 6548 +6 6C6C
001A	RET Far	CB	(AX) = 0905 (DX) = 0010 (DS) = 1A05 (SP) = FFFC (IP) = 001A (CS) = A108 Stack +0 0000 +2 19F5 +4 6548 +6 6C6C	(AX) = 0905 (DX) = 0010 (DS) = 1A05 (SP) = 0000 (IP) = 0000 (CS) = 19F5 Stack +0 6548 +2 6C6C +4 206F +6 6F57

0000	INT 20	CD20	(AX) = 0905 (DX) = 0010 (DS) = 1A05 (SP) = 0000 (IP) = 0000 Stack +0 6548 +2 6C6C +4 206F +6 6F57	(AX) = 0000 (DX) = 0000 (DS) = 19F5 (SP) = 0000 (IP) = 0005 Stack +0 6548 +2 6C6C +4 206F +6 6F57
------	--------	------	---	---

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*

```
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.9382 - Юрьев
Сергей.',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*

```
EOFLine EQU '$'
AStack SEGMENT STACK
DW 12 DUP(?)
AStack ENDS
DATA SEGMENT
HELLO DB 'Hello Worlds!', 0AH, 0DH, EOFLine
GREETING DB 'Yuryev Sergey from 9382.', 0AH, 0DH, EOFLine
DATA ENDS
```

```

CODE      SEGMENT
          ASSUME CS:Code DS:DATA SS:STACK
WriteMsg  PROC    NEAR
          mov     AH,9
          int     21h
          ret
WriteMsg  ENDP
Main      PROC    FAR
          push    DS
          sub     AX,AX
          push    AX
          mov     AX,DATA
          mov     DS,AX
          mov     DX, OFFSET HELLO
          call    WriteMsg
          mov     DX, OFFSET GREETING
          call    WriteMsg
          ret
Main      ENDP
CODE      ENDS
          END Main

```