

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. 9382

Савельев И.С.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

Цель работы.

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

Ход работы.

Часть 1.

Загружены файлы hello1.asm, hello2.asm, masm.exe, link.exe, afd.com, lib.exe из каталога \лабораторные_работы в каталог /home/indiora/1.

Запущена программа DOSBox, смонтирован виртуальный диск D: при помощи mount D /home/indiora/1.

Преобразована строка-приветствие в соответствии с личными данными.

В DOS осуществлен переход на виртуальный диск при помощи команды D:

Загружена русская кодовая таблица символов путем набора строки:

```
> keyb ru 866
```

Протранслирована программа с помощью строки:

```
> masm hello1.asm
```

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

Скомпонован загрузочный модуль Hello1.exe с помощью строки:

```
> link Hello1.obj
```

В результате работы линковщика создается загрузочный модуль Hello1.exe

Запущена программа в автоматическом режиме путем набора строки:

```
> Hello1.exe
```

Вывод программы:

```
> Вас приветствует ст.гр.9382 — Савельев Илья.
```

Выполнен запуск программы Hello1.exe в пошаговом режиме с фиксацией используемых регистров и ячеек памяти до и после выполнения каждой команды в таблице 1, используя отладчик и соответственно команду:

```
> afdpro Hello1.exe.
```

Таблица 1.

Содержимое сегментных регистров до старта программы: CS:1A05, DS:19F5, ES:19F5, SS:1A0A, HS:19F5, FS:19F5, DI: 0000, SI: 0000.

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 (CX) = 004C	(AX) = 1A07 (DS) = 19F5 (DX) = 0000 (IP) = 0013 (CX) = 004C
0013	MOV DS, AX	8ED8	(AX) = 1A07 (DS) = 19F5 (DX) = 0000 (IP) = 0013 (CX) = 004C	(AX) = 1A07 (DS) = 1A07 (DX) = 0000 (IP) = 0015 (CX) = 004C
0015	MOV DX, 0000	BA0000	(AX) = 1A07 (DS) = 1A07 (DX) = 0000 (IP) = 0015 (CX) = 004C	(AX) = 1A07 (DS) = 1A07 (DX) = 0000 (IP) = 0018 (CX) = 004C
0018	MOV AH,09	B409	(AX) = 1A07 (DS) = 1A07 (DX) = 0000 (IP) = 0018 (CX) = 004C	(AX) = 0907 (DS) = 1A07 (DX) = 0000 (IP) = 001A (CX) = 004C
001A	INT 21	CD21	(AX) = 0907 (DS) = 1A07 (DX) = 0000 (IP) = 001A (CX) = 004C	(AX) = 0907 (DS) = 1A07 (DX) = 0000 (IP) = 001C (CX) = 004C
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 (CX) = 004C	(AX) = 0000 (DS) = 19F5 (DX) = 0000 (IP) = 0010 (CX) = 0000

Часть 2.

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

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

```
>masm Hello2.asm
```

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

Используя линковщик LINK, создан загрузочный модуль Hello2.exe.

Использована команда:

```
>link Hello2.obj
```

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

```
>Hello Worlds!
```

```
>Savelyev I.S. - from 9382.
```

Запущена программа Hello2.exe в пошаговом режиме, используя отладчик afdpro с фиксацией содержимого используемых регистров и ячеек памяти до и после выполнения команд в таблице 2.

Таблица 2

Содержимое сегментных регистров до старта программы: CS:1A0A, DS:19F5, ES:19F5, SS:1A05, HS:19F5, FS: 19F5, SI: 0000, DI: 0000, BP: 0000.

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 (CX) = 006B (CS) = 1A0A	(AX) = 0000 (DS) = 19F5 (SP) = 0016 (IP) = 0006 (CX) = 006B (CS) = 1A0A

			Stack +0 0000 +2 0000 +4 0000	Stack +0 19F5 +2 0000 +4 0000
0006	SUB AX, AX	2BC0	(AX) = 0000 (DX) = 0000 (DS) = 19F5 (SP) = 0016 (IP) = 0006 (CX) = 006B (CS) = 1A0A Stack +0 19F5 +2 0000 +4 0000	(AX) = 0000 (DX) = 0000 (DS) = 19F5 (SP) = 0016 (IP) = 0008 (CX) = 006B (CS) = 1A0A Stack +0 19F5 +2 0000 +4 0000
0008	PUSH AX	50	(AX) = 0000 (DX) = 0000 (DS) = 19F5 (SP) = 0016 (IP) = 0008 (CX) = 006B (CS) = 1A0A Stack +0 19F5 +2 0000 +4 0000	(AX) = 0000 (DX) = 0000 (DS) = 19F5 (SP) = 0014 (IP) = 0009 (CX) = 006B (CS) = 1A0A Stack +0 0000 +2 19F5 +4 0000
0009	MOV AX, 1A07	B8071A	(AX) = 0000 (DX) = 0000 (DS) = 19F5 (SP) = 0014 (IP) = 0009 (CX) = 006B (CS) = 1A0A Stack +0 0000 +2 19F5 +4 0000	(AX) = 1A07 (DX) = 0000 (DS) = 19F5 (SP) = 0014 (IP) = 000C (CX) = 006B (CS) = 1A0A Stack +0 0000 +2 19F5 +4 0000
000C	MOV DS, AX	8ED8	(AX) = 1A07 (DX) = 0000 (DS) = 19F5 (SP) = 0014 (IP) = 000C (CX) = 006B (CS) = 1A0A Stack +0 0000 +2 19F5 +4 0000	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (SP) = 0014 (IP) = 000E (CX) = 006B (CS) = 1A0A Stack +0 0000 +2 19F5 +4 0000
000E	MOV DX, 0000	BA0000	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (SP) = 0014 (IP) = 000E	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (SP) = 0014 (IP) = 0011

			(CX) = 006B (CS) = 1A0A Stack +0 0000 +2 19F5 +4 0000	(CX) = 006B (CS) = 1A0A Stack +0 0000 +2 19F5 +4 0000
0011	CALL 0000	E8ECFF	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (SP) = 0014 (IP) = 0011 (CX) = 006B (CS) = 1A0A Stack +0 0000 +2 19F5 +4 0000	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (SP) = 0012 (IP) = 0000 (CX) = 006B (CS) = 1A0A Stack +0 0014 +2 0000 +4 19F5
0000	MOV AH,09	B409	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (SP) = 0012 (IP) = 0000 (CX) = 006B (CS) = 1A0A Stack +0 0014 +2 0000 +4 19F5	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (SP) = 0012 (IP) = 0002 (CX) = 006B (CS) = 1A0A Stack +0 0014 +2 0000 +4 19F5
0002	INT 21	CD21	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (SP) = 0012 (IP) = 0002 (CX) = 006B (CS) = 1A0A Stack +0 0014 +2 0000 +4 19F5	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (SP) = 0012 (IP) = 0004 (CX) = 006B (CS) = 1A0A Stack +0 0014 +2 0000 +4 19F5
0004	RET	C3	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (SP) = 0012 (IP) = 0004 (CX) = 006B (CS) = 1A0A Stack +0 0014 +2 0000 +4 19F5	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (SP) = 0014 (IP) = 0014 (CX) = 006B (CS) = 1A0A Stack +0 0000 +2 19F5 +4 0000
0014	MOV DX,0010	BA1000	(AX) = 0907 (DX) = 0000 (DS) = 1A07	(AX) = 0907 (DX) = 0010 (DS) = 1A07

			(SP) = 0014 (IP) = 0014 (CX) = 006B (CS) = 1A0A Stack +0 0000 +2 19F5 +4 0000	(SP) = 0014 (IP) = 0017 (CX) = 006B (CS) = 1A0A Stack +0 0000 +2 19F5 +4 0000
0017	CALL 0000	E8E6FF	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (SP) = 0014 (IP) = 0017 (CX) = 006B (CS) = 1A0A Stack +0 0000 +2 19F5 +4 0000	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (SP) = 0012 (IP) = 0000 (CX) = 006B (CS) = 1A0A Stack +0 001A +2 0000 +4 19F5
0000	MOV AH,09	B409	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (SP) = 0012 (IP) = 0000 (CX) = 006B (CS) = 1A0A Stack +0 001A +2 0000 +4 19F5	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (SP) = 0012 (IP) = 0002 (CX) = 006B (CS) = 1A0A Stack +0 001A +2 0000 +4 19F5
0002	INT 21	CD21	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (SP) = 0012 (IP) = 0002 (CX) = 006B (CS) = 1A0A Stack +0 001A +2 0000 +4 19F5	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (SP) = 0012 (IP) = 0004 (CX) = 006B (CS) = 1A0A Stack +0 001A +2 0000 +4 19F5
0004	RET	C3	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (SP) = 0012 (IP) = 0004 (CX) = 006B (CS) = 1A0A Stack +0 001A +2 0000 +4 19F5	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (SP) = 0014 (IP) = 001A (CX) = 006B (CS) = 1A0A Stack +0 0000 +2 19F5 +4 0000

001A	RET Far	CB	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (SP) = 0014 (IP) = 001A (CX) = 006B (CS) = 1A0A Stack +0 0000 +2 19F5 +4 0000	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (SP) = 0018 (IP) = 0000 (CX) = 006B (CS) = 19F5 Stack +0 0000 +2 0000 +4 0000
0000	INT 20	CD20	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (SP) = 0018 (IP) = 0000 (CX) = 006B (CS) = 19F5 Stack +0 0000 +2 0000 +4 0000	(AX) = 0000 (DX) = 0000 (DS) = 19F5 (SP) = 0018 (IP) = 0005 (CX) = 0000 (CS) = 1A0A Stack +0 0000 +2 0000 +4 0000

Выводы.

В результате выполнения лабораторной работы была освоена трансляция и отладка программ на языке Ассемблера.

Приложение А. Исходный код программы.

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 - Савельев И.С.',13,10,'$'
.CODE ;Начало сегмента кода
mov ax, @data ;Загрузка в DS адреса
mov ds, ax ;начала 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 *hello1.lst*

```
1
2 ; HELLO1.ASM - упрощенная версия учебн
ой программы лаб.раб. N1
3 ; по дисциплине "Архитектура компьютера"
4 ; *****
*****
5 ; Назначение: Программа формирует и выв
одит на экран приветствие
6 ; пользователь с помощью фу
нкции ДОС "Вывод строки"
7 ; (номер 09 прерывание 21h)
, которая:
8 ; - обеспечивает вывод на
экран строки символов,
9 ; заканчивающейся знаком
"$";
10 ; - требует задания в реги
стре ah номера функции=09h,
11 ; а в регистре dx - сме
щения адреса выводимой
12 ; строки;
13 ; - использует регистр ax
и не сохраняет его
14 ; содержимое.
15 ; *****
*****
16
17 DOSSEG
; Задание сегментов под ДОС
18 .MODEL SMALL
; Модель памяти-SMALL (Малая)
19 .STACK 100h
; Отвести под стек 256 байт
20 .DATA
; Начало сегмента данных
21 0000 Greeting LABEL BYTE
; Текст приветствия
22 0000 82 A0 E1 20 AF E0 DB 'Вас приветствует ст.гр.7303 - Ив
анов И.И.',13,10,'$'
23 A8 A2 A5 E2 E1 E2
```

```

24      A2 E3 A5 E2 20 E1
25      E2 2E A3 E0 2E 37
26      33 30 33 20 2D 20
27      88 A2 A0 AD AE A2
28      20 88 2E 88 2E 0D
29      0A 24
30                                     .CODE
                                     ; Начало сегмента кода
31 0000  B8 ---- R                   mov ax, @data
                                     ; Загрузка в DS адреса начала
32 0003  8E D8                       mov ds, ax
                                     ; сегмента данных
33 0005  BA 0000 R                   mov dx, OFFSET Greeting
Microsoft (R) Macro Assembler Version 5.10          9/23/20 22:44:21
                                                    Page 1-2

```

```

                                     ; Загрузка в dx смещения
34
                                     ; адреса текста приветствия
35 0008                                DisplayGreeting:
36 0008  B4 09                       mov ah, 9
                                     ; # функции ДОС печати строки
37 000A  CD 21                       int 21h
                                     ; вывод на экран приветствия
38 000C  B4 4C                       mov ah, 4ch
                                     ; # функции ДОС завершения программы
39 000E  CD 21                       int 21h
                                     ; завершение программы и выход в ДОС
40                                END
Microsoft (R) Macro Assembler Version 5.10          9/23/20 22:44:21
                                                    Symbols-1

```

Segments and Groups:

N a m e	Length	Align	Combine	Class
DGROUP	GROUP			
_DATA	002C	WORD	PUBLIC	'DATA'
_STACK	0100	PARA	STACK	'STACK'
_TEXT	0010	WORD	PUBLIC	'CODE'

Symbols:

N a m e	Type	Value	Attr
DISPLAYGREETING	L NEAR	0008	_TEXT
GREETING	L BYTE	0000	_DATA
@CODE	TEXT	_TEXT	
@CODESIZE	TEXT	0	
@CPU	TEXT	0101h	
@DATASIZE	TEXT	0	
@FILENAME	TEXT	hello1	
@VERSION	TEXT	510	

```

33 Source  Lines
33 Total   Lines
19 Symbols

```

47476 + 461831 Bytes symbol space free

0 Warning Errors
0 Severe Errors

Текст файла hello2.asm

```
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 9382 - Savelyev I.S.$'
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
    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 1-ом слове PSP.
Main ENDP
CODE ENDS
END Main
```

Текст файла hello2.lst

```
1 ; HELLO2 - Учебная программа N2 лаб.ра
; б. #1 по дисциплине "Архитектура компьют
; ера"
2 ; Программа использует процеду
; ру для печати строки
3 ;
4 ; ТЕКСТ ПРОГРАММЫ
5
6 = 0024 EOFLine EQU '$' ; Определение
; символьной константы
7 ; "Конец
; строки"
8
9 ; Стек программы
10
11 0000 AStack SEGMENT STACK
12 0000 000C[ DW 12 DUP(?) ; Отводится 1
; 2 слов памяти
```

```

13      ????
14      ]
15
16 0018      AStack      ENDS
17
18      ; Данные программы
19
20 0000      DATA      SEGMENT
21
22      ; Директивы описания данных
23
24 0000  48 65 6C 6C 6F 20      HELLO  DB 'Hello Worlds!', 0AH, 0DH,
      EOFLine
25      57 6F 72 6C 64 73
26      21 0A 0D 24
27 0010  53 74 75 64 65 6E      GREETING  DB 'Student from 4350 - $'
28      74 20 66 72 6F 6D
29      20 34 33 35 30 20
30      2D 20 24
31 0025      DATA      ENDS
32
33      ; Код программы
34
35 0000      CODE      SEGMENT
36      ASSUME CS:Code DS:DATA SS:ASt
      ack
hello2.asm(28): warning A4001: Extra characters on line
37      ; Процедура печати строки
38 0000      WriteMsg  PROC  NEAR
39 0000  B4 09      mov  AH,9
40 0002  CD 21      int  21h ; Вызов функции DO
      S по прерыванию
41 0004  C3      ret
42 0005      WriteMsg  ENDP
43
44      ; Головная процедура
45 0005      Main      PROC  FAR
Microsoft (R) Macro Assembler Version 5.10
9/24/20 11:12:18
Page 1-2

46 0005  1E      push  DS      ;\ Сохранение
      адреса начала PSP в стеке
47 0006  2B C0      sub  AX,AX ; > для послед
      ующего восстановления по
48 0008  50      push  AX      ;/ команде re
      t, завершающей процедуру.
49 0009  B8 ---- R  mov  AX,DATA      ; 3
      агрузка сегментного
50 000C  8E D8      mov  DS,AX      ; p
      егистра данных.
51 000E  BA 0000 R  mov  DX, OFFSET HELLO ; B
      ывод на экран первой
52 0011  E8 0000 R  call WriteMsg      ; c
      троки приветствия.
53 0014  BA 0010 R  mov  DX, OFFSET GREETING ; B
      ывод на экран второй
54 0017  E8 0000 R  call WriteMsg      ; c
      троки приветствия.
55 001A  CB      ret      ; B
      ыход в DOS по команде,
56      ; н
      аходящейся в 1-ом слове PSP.
57 001B      Main      ENDP

```

```

58 001B          CODE      ENDS
59              END Main
Microsoft (R) Macro Assembler Version 5.10
9/24/20 11:12:18
Symbols-1

```

Segments and Groups:

N a m e	Length	Align	Combine Class
ASTACK	0018	PARA	STACK
CODE	001B	PARA	NONE
DATA	0025	PARA	NONE

Symbols:

N a m e	Type	Value	Attr	
EOFLINE	NUMBER	0024		
GREETING	L BYTE	0010	DATA	
HELLO	L BYTE	0000	DATA	
MAIN	F PROC	0005	CODE	Length = 0016
WRITEMSG	N PROC	0000	CODE	Length = 0005
@CPU	TEXT	0101h		
@FILENAME	TEXT	hello2		
@VERSION	TEXT	510		

```

51 Source Lines
51 Total Lines
13 Symbols

```

47484 + 461823 Bytes symbol space free

```

0 Warning Errors
0 Severe Errors

```