

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

Михайлов Д.А.

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. Загрузил файлы hello1.asm, hello2.asm, masm.exe, link.exe, afdpro.com, lib.exe из каталога \лаборат_работы в каталог c:\MASM.

2. Запустил программу DOSBox, смонтировал виртуальный диск c: в каталоге \MASM при помощи mount c:\MASM.

3. Просмотрел программу в режиме редактирования, изучил структуру и реализации каждого сегмента программы. Строку-приветствие преобразовал в соответствии со своими личными данными.

4. В DOS перешёл на виртуальный диск при помощи команды c:

5. Протранслировал программу с помощью строки:

> masm hello1.asm

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

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

> link Hello1.obj

В результате работы линковщика создается загрузочный модуль Hello1.exe.

7. Загрузил русскую кодовую таблицу символов путём набора строки:

> keyb ru 866

8. Запустил программу в автоматическом режиме путем набора строки:

> Hello1.exe

9. Вывод программы:

> Вас приветствует ст.гр.9382 — Михайлов Д.А..

10. Выполнил 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:

> afd 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. Просмотрел программу Hello2.asm в режиме редактирования, изучил ее структуру и реализацию каждого сегмента программы. Строки-приветствия преобразовывал в соответствии со своими личными данными.

2. Выполнил трансляцию 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, создал загрузочный модуль Hello2.exe. Использовал команду:

>link Hello2.obj

4. Выполнил программу Hello2.exe в автоматическом режиме и убедился в том, что она работает корректно: в консоль выводится:

>Hello worlds!

>Greetings from student of 9382 Mikhailov Dmitry.

5. Запустил программу Hello2.exe в пошаговом режиме, используя отладчик afd с фиксацией содержимого используемых регистров и ячеек памяти до и после выполнения команд.

Содержимое сегментных регистров до старта программы: CS:11B2, DS:119C, ES:119C, SS:11AC, HS:119C, FS:119C

Табл.2.

Адрес Команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0005	PUSH DS	1E	(DS) = 119C (SP)=0018 (IP) = 0005 Stack +0 0000 +2 0000 +4 0000	(DS) = 119C (SP)=0016 (IP) = 0006 Stack +0 119C +2 0000 +4 0000
0006	SUB AX, AX	2BC0	(AX) = 0000 (IP) = 0006	(AX) = 0000 (IP) = 0008

0008	PUSH AX	50	(AX) = 0000 (SP)=0016 (IP) = 0008 Stack +0 119C +2 0000 +4 0000	(AX) = 0000 (SP)=0014 (IP) = 0009 Stack +0 0000 +2 119C +4 0000
0009	MOV AX, 11AE	B8AE11	(AX) = 0000 (IP) = 0009	(AX) = 11AE (IP) = 000C
000C	MOV DS, AX	8ED8	(AX) = 11AE (DS) = 119C (IP) = 000C	(AX) = 11AE (DS) = 11AE (IP) = 000E
000E	MOV DX, 0000	BA0000	(DX) = 0000 (IP) = 000E	(DX) = 0000 (IP) = 0011
0011	CALL 0000	E8ECFF	(SP)=0014 (IP) = 0011 Stack +0 0000 +2 119C +4 0000	(SP)=0012 (IP) = 0000 Stack +0 0014 +2 0000 +4 119C
0000	MOV AH,09	B409	(AX) = 11AE (IP) = 0000	(AX) = 09AE (IP) = 0002
0002	INT 21	CD21	(IP) = 0002	(IP) = 0004
0004	RET	C3	(SP)=0012 (IP) = 0004 Stack +0 0014 +2 0000 +4 119C	(SP)=0014 (IP) = 0014 Stack +0 0000 +2 119C +4 0000
0014	MOV DX,0010	BA1000	(DX) = 0000 (IP) = 0014	(DX) = 0010 (IP) = 0017
0017	CALL 0000	E8E6FF	(SP)=0014 (IP) = 0017 Stack +0 0000 +2 119C +4 0000	(SP)=0012 (IP) = 0000 Stack +0 001A +2 0000 +4 119C
0000	MOV AH,09	B409	(AX) = 09AE (IP) = 0000	(AX) = 09AE (IP) = 0002
0002	INT 21	CD21	(IP) = 0002	(IP) = 0004
0004	RET	C3	(SP)=0012 (IP) = 0004 Stack +0 001A +2 0000 +4 119C	(SP)=0014 (IP) = 001A Stack +0 0000 +2 119C +4 0000
001A	RET Far	CB	(SP)=0014 (IP) = 001A Stack +0 0000 +2 119C +4 0000	(SP)=0018 (IP) = 0000 Stack +0 0000 +2 0000 +4 0000

0000	INT 20	CD20	(AX) = 09AE (DS) = 11AE (IP) = 0000	(AX) = 0000 (DS) = 119C (IP) = 0005
------	--------	------	---	---

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 Михайлов
Дмитрий',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 'Здравствуйте!', 0AH, 0DH, EOFLine
GREETING DB 'Вас приветствует ст.гр.9382 - Михайлов
Дмитрий.', 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

```

Текст файла hello1.lst

```

#Microsoft      (R)      Macro      Assembler      Version      5.10
10/22/20 12:56:0

```

Pag

e 1-1

```

прогр          ; HELLO1.ASM -      упрощенная версия учебной
               аммы лаб.раб. N1
               ;                  по дисциплине "Архитектура
комп           ьютера"
               ;
*****
               ; Назначение: Программа формирует и выводит
на             экран приветствие
               ;                  пользователя с помощью функции
до

```



```

; С "Вывод строки"
; (номер 09 прерывание 21h),
котора
;
; я:
; - обеспечивает вывод на экран
;
; строки символов,
; заканчивающейся знаком "$";
; - требует задания в регистре
ah
; номера функции=09h,
; а в регистре dx - смещения
а
; адреса выводимой
; строки;
; - использует регистр ax и
не
; сохраняет его
; содержимое.
;
*****
*****

```

```

DOSSEG
; Задание сегментов под DOS
.MODEL SMALL
; Модель памяти-SMALL(Малая)
.STACK 100h
; Отвести под Стек 256 байт
.DATA
; Начало сегмента данных
0000 Greeting LABEL BYTE
; Текст приветствия
0000 82 A0 E1 20 AF E0 DB 'Вас приветствует ст.гр.9382
Михайлов Дми
; трий',13,10,'$'
A8 A2 A5 E2 E1 E2
A2 E3 A5 E2 20 E1
E2 2E A3 E0 2E 39
33 38 32 20 8C A8
E5 A0 A9 AB AE A2
20 84 AC A8 E2 E0
A8 A9 0D 0A 24
.CODE
;
Начал
о сегмента кода

```

```

0000 B8 ---- R                                mov          ax,      @data
; Загр                                         узка в DS адреса начала
0003 8E D8                                mov          ds,      ax
; сегм                                         ента данных
0005 BA 0000 R                                mov          dx,      OFFSET Greeting
; Загр                                         узка в dx смещения

```

```

#Microsoft      (R)      Macro      Assembler      Version      5.10
10/22/20 12:56:0
Pag
e      1-2

```

```

;
адрес
а текста приветствия
0008      DisplayGreeting:
0008 B4 09                                mov          ah,      9
; # фy                                         нкции ДОС печати строки
000A CD 21                                int
21h                                           ; вывод
на экран приветствия
000C B4 4C                                mov          ah,      4ch
; # фy                                         нкции ДОС завершения программы
000E CD 21                                int
21h                                           ; завер
шение программы и выход в ДОС
END

```

```

#Microsoft      (R)      Macro      Assembler      Version      5.10
10/22/20 12:56:0
Sym
bols-1

```

Segments and Groups:

	N a m e	Length	Align	Combine
Class				

DGROUP	GROUP		
_DATA	002F 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

47998 + 461309 Bytes symbol space free

0 Warning Errors
 0 Severe Errors

Текст файла hello1.2st

#Microsoft (R) Macro Assembler Version 5.10
10/22/20 12:56:3

Pag

e 1-1

по ; HELLO2 - Учебная программа N2 лаб.раб.#1
дисциплине "Архитектура компьютера"
; Программа использует процедуру для
п ечати строки
;
; ТЕКСТ ПРОГРАММЫ
= 0024 EOFLine EQU '\$' ; Определение
СИМВОЛЬ ной константы
; "Конец
строки"

; Стек программы
0000 AStack SEGMENT STACK
0000 000C[DW 12 DUP(?) ; Отводится 12
слов п амяти
????]

0018 AStack ENDS

; Данные программы

0000 DATA SEGMENT

; Директивы описания данных

0000 48 65 6C 6C 6F 20 HELLO DB 'Hello Worlds!', 0AH,
0DH, EOFLine
57 6F 72 6C 64 73
21 0A 0D 24
0010 47 72 65 65 74 69 GREETING DB 'Greetings from Mikhailov
Dmitry f

```

                                rom 9382 '$'
6E 67 73 20 66 72
6F 6D 20 4D 69 6B
68 61 69 6C 6F 76
20 44 6D 69 74 72
79 20 66 72 6F 6D
20 39 33 38 32 20
24
003B                                DATA            ENDS

                                ; Код программы

0000                                CODE            SEGMENT
                                ASSUME CS:Code DS:DATA SS:AStack
hello2.ASM(28): warning A4001: Extra characters on line
                                ; Процедура печати строки
0000                                WriteMsg  PROC  NEAR
0000  B4 09                                mov    AH,9
0002  CD 21                                int     21h  ; Вызов функции DOS
по пре
                                рыванию
0004  C3                                ret
0005                                WriteMsg  ENDP

                                ; Головная процедура

```

e 1-2

```

0005          Main      PROC  FAR
0005  1E          push  DS          ;\  Сохранение
адреса
                                начала PSP в стеке
0006  2B C0          sub   AX,AX      ; > для
последующего в
                                останова по
0008  50          push  AX          ;/  команде ret,
завер
                                шающей процедуру.
0009  B8 ---- R      mov   AX,DATA      ;
Загрузка
                                сегментного
000C  8E D8          mov   DS,AX      ;
регистра
                                данных.
000E  BA 0000 R      mov   DX, OFFSET HELLO ;
Вывод на
                                экран первой
0011  E8 0000 R      call  WriteMsg      ;
строки пр
                                иветствия.
0014  BA 0010 R      mov   DX, OFFSET GREETING ;
Вывод на
                                экран второй
0017  E8 0000 R      call  WriteMsg      ;
строки пр
                                иветствия.
001A  CB          ret              ; Выход в
D
                                OS по команде,
                                ;
находящей
                                ся в 1-ом слове PSP.
001B          Main      ENDP
001B          CODE      ENDS
                                END Main

```

#Microsoft (R) Macro Assembler Version 5.10
10/22/20 12:56:3

Sym

bols-1

Segments and Groups:

Class	N a m e	Length	Align	Combine
ASTACK	0018	PARA	STACK
CODE	001B	PARA	NONE
DATA	003B	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

48006 + 461301 Bytes symbol space free

1 Warning Errors
0 Severe Errors