

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

ОТЧЕТ
по лабораторной работе №1
по дисциплине «Организация ЭВМ и систем»
Тема: Трансляция, отладка и выполнение программ на языке
Ассемблера

Студент гр. _____ Беззубов Д.В.

Преподаватель _____ Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

Ознакомиться с трансляцией, выполнением и отладкой программы на языке Ассемблер.

Ход работы.

Часть 1.

1. В программе DOSBOX была подключена папка MASM с исходными файлами.
2. Программа hello1.asm была просмотрена в режиме редактирования и была преобразована строка приветствия.
3. Программа была транслирована с помощью команды:

```
> masm hello1.asm
```

И были созданы объектный файл hello1.obj и файл листинга hello1.lst, которые приведены в приложении А.

4. Был скомпонован загрузочный модуль hello1.exe с помощью команды:

```
> link hello1.obj
```

5. Программа была выполнена в автоматическом режиме с помощью команды:

```
> hello1.exe
```



```
C:\>HELLO1.EXE
Вас приветствует ст.гр.1303 - Беззубов Д.В.
```

Рисунок 1 – запуск hello1.exe

Верное выполнение программы подтверждается рисунком.

6. Программа hello1 была выполнена в пошаговом режиме под управлением отладчика отладчика afd:

```
> afd hello1.exe
```

Стартовые значения регистров:

- DS = 19F5
- SS = 1A0A
- ES = 19F5

- CS = 1A05

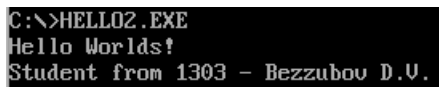
Результат выполнения программы представлен в таблице 1.

Таблица 1

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 IP = 0010	AX = 1A07 DS = 19F5 IP = 0013
0013	MOV DS, AX	8ED8	AX = 1A07 DS = 19F5 IP = 0013	AX = 1A07 DS = 1A07 IP = 0015
0015	MOV DX, 0000	BA0000	AX = 1A07 DS = 1A07 IP = 0015	AX = 1A07 DS = 1A07 IP = 0018
0018	MOV AH, 09	B409	AX = 1A07 DS = 1A07 IP = 0018	AX = 0907 DS = 1A07 IP = 001A
001A	INT 21	CD21	AX = 0907 DS = 1A07 IP = 001A	AX = 0907 DS = 1A07 IP = 001C
001C	MOV AH, 4C	B44C	AX = 0907 DS = 1A07 IP = 001C	AX = 4C07 DS = 1A07 IP = 001E
001E	INT 21	CD21	AX = 4C07 DS = 1A07 IP = 001E	AX = 0000 DS = 19F5 IP = 0010

Часть 2

1. Программа hello2.asm была просмотрена в режиме редактирования. Была отредактирована строка приветствия. Листинг программы приложен в приложении В.
2. Программа hello2.asm была транслирована для создания объектного файла hello2.obj и файла листинга hello2.lst, листинг представлен в приложении В.
3. Программа hello2.exe была скомпонована с помощью компоновщика Link.
4. Был выполнен запуск программы и проверка на её работоспособность.



```
C:\>HELLO2.EXE
Hello Worlds!
Student from 1303 - Bezzubov D.U.

```

Рисунок 3 – запуск hello2.exe

5. Программа hello2 была выполнена в пошаговом режиме под управлением отладчика отладчика afd:

> afd hello2.exe

Стартовые значения регистров:

- DS = 19F5
- SS = 1A05
- ES = 19F5
- CS = 1A0A

Результат выполнения программы представлен в таблице 2.

Таблица 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 IP = 0005 SP = 0018 STACK +0 = 0000	AX = 0000 DS = 19F5 IP = 0006 SP = 0016 STACK +0 = 19F5

0006	SUB AX, AX	2BC0	AX = 0000 DS = 19F5 IP = 0006	AX = 0000 DS = 19F5 IP = 0008
0008	PUSH AX	50	AX = 0000 DS = 19F5 IP = 0008 SP = 0016 STACK +0 = 19F5 STACK +2 = 0000	AX = 0000 DS = 19F5 IP = 0009 SP = 0014 STACK +0 = 0000 STACK +2 = 19F5
0009	MOV AX, 1A07	B8071A	AX = 0000 IP = 0009 SP = 0014	AX = 1A07 IP = 000C SP = 0014
000C	MOV DS, AX	8ED8	AX = 1A07 DS = 19F5 IP = 000C	AX = 1A07 DS = 1A07 IP = 000E
000E	MOV DX, 0000	BA0000	AX = 1A07 DS = 1A07 IP = 000E	AX = 1A07 DS = 1A07 IP = 0011
0011	CALL 0000	E8ECFF	AX = 1A07 DS = 1A07 IP = 0011 SP = 0014	AX = 1A07 DS = 1A07 IP = 0000 SP = 0012
0000	MOV AH, 09	B409	AX = 1A07 IP = 0000 STACK +0 = 0000 STACK +2 = 19F5 STACK +4 = 0000	AX = 0907 IP = 0002 STACK +0 = 0014 STACK +2 = 0000 STACK +4 = 19F5
0002	INT 21	CD21	AX = 0907 DS = 1A07 IP = 0002	AX = 0907 DS = 1A07 IP = 0004
0004	RET	C3	IP = 0004 SP = 0012 STACK +0 = 0014 STACK +2 = 0000 STACK +4 = 19F5	IP = 0014 SP = 0014 STACK +0 = 0000 STACK +2 = 19F5 STACK +4 = 0000
0014	MOV DX, 0010	BA1000	AX = 0907 DS = 1A07 IP = 0014 DX = 0000	AX = 0907 DS = 1A07 IP = 0017 DX = 0010
0017	CALL 0000	E8E6FF	AX = 0907 DS = 1A07 SP = 0014 IP = 0017 DX = 0010	AX = 0907 DS = 1A07 SP = 0012 IP = 0000 DX = 0010

			STACK +0 = 0000 STACK +2 = 19F5 STACK +4 = 0000	STACK +0 = 001A STACK +2 = 0000 STACK +4 = 19F5
0000	MOV AH, 09	B409	AX = 0907 DS = 1A07 IP = 0000	AX = 0907 DS = 1A07 IP = 0002
0002	INT 21	CD21	AX = 0907 DS = 1A07 IP = 0002	AX = 0907 DS = 1A07 IP = 0004
0004	RET	C3	AX = 0907 DS = 1A07 SP = 0012 IP = 0004 STACK +0 = 001A STACK +2 = 0000 STACK +4 = 19F5	AX = 0907 DS = 1A07 SP = 0014 IP = 001A STACK +0 = 0000 STACK +2 = 19F5 STACK +4 = 0000
001A	RET	CB	AX = 0907 DS = 1A07 CS = 1A0B SP = 0014 IP = 001A STACK +0 = 0000 STACK +2 = 19F5	AX = 0907 DS = 1A07 CS = 19F5 SP = 0018 IP = 0000 STACK +0 = 0000 STACK +2 = 0000
0000	INT 20	CD20	AX = 0907 DS = 1A07 SP = 0018	AX = 0000 DS = 19F5 SP = 0018

Вывод:

В ходе лабораторной работы было произведена трансляция, линковка, выполнение и отладка программ на языке Ассемблера. Результатом выполнения лабораторной работы являются рабочие программы hello1.exe и hello2.exe, а также протокол, в котором отражены изменения состояний регистров памяти в ходе выполнения программ.

ПРИЛОЖЕНИЕ А

Код программы hello1.asm

```

DOSSEG
.MODEL    SMALL
.STACK    100h
.DATA
Greeting LABEL BYTE
    DB 'Вас приветствует ст.гр.1303 - Беззубов Д.В.',13,10,'$'
.CODE
    mov ax, @data
    mov ds, ax
    mov dx, OFFSET Greeting

DisplayGreeting:
    mov ah, 9
    int 21h
    mov ah, 4ch
    int 21h
END

```

Файл hello1.lst

```

Microsoft      (R)      Macro      Assembler      Version      5.10
9/16/22 22:55:58

```

Page 1-1

```

                                DOSSEG
                                .MODEL    SMALL
                                .STACK    100h
                                .DATA
0000                                Greeting LABEL BYTE
0000 82 A0 E1 20 AF E0      DB 'Вас приветствует ст.гр.1303 -
Беззубов Д.В.',13,10,'$'
                                A8 A2 A5 E2 E1 E2
                                A2 E3 A5 E2 20 E1
                                E2 2E A3 E0 2E 31
                                33 30 33 20 2D 20
                                81 A5 A7 A7 E3 A1
                                AE A2 20 84 2E 82
                                2E 0D 0A 24
                                .CODE
0000 B8 ---- R      mov ax, @data
0003 8E D8      mov ds, ax
0005 BA 0000 R      mov dx, OFFSET Greeting

```

Page 1-2

```

0008                                DisplayGreeting:
0008 B4 09                          mov ah, 9
000A CD 21                          int 21h
000C B4 4C                          mov ah, 4ch
000E CD 21                          int 21h
                                END

```

Symbols-1

Segments and Groups:

	N a m e	Length	Align
Combine Class			
DGROUP	GROUP	
_DATA	002E 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

47994 + 461313 Bytes symbol space free

0 Warning Errors

0 Severe Errors

ПРИЛОЖЕНИЕ Б

Код программы 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 - Bezzubov D.V.$'
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

```

Файл hello2.lst

```

Microsoft      (R)      Macro      Assembler      Version      5.10
9/16/22 22:58:38

```

Page 1-1

```

; HELLO2 - Учебная программа N2
; лаб.раб.#1 по дисциплине "Архитектура компьютера"
; Программа использует процедуру для печати строки
;
; ТЕКСТ ПРОГРАММЫ

= 0024      EOFLine EQU '$'      ; Определены символы строки
; "Конец строки"

; стек программы

ASSUME CS:CODE, SS:AStack

0000      AStack      SEGMENT      STACK
0000 000C[      DW      12      DUP('!')      ;
Отводит 12 слов памяти
0021
]

0018      AStack      ENDS

; Данные программы

```

```

0000                                DATA        SEGMENT

                                ;  Директивы описания данн
                                ых

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  53 74 75 64 65 6E GREETING  DB 'Student from 4350 - $'
                                74 20 66 72 6F 6D
                                20 34 33 35 30 20
                                2D 20 24
0025                                DATA        ENDS

                                ;  Код программы

0000                                CODE        SEGMENT

                                ;  Процедура печати строки
0000                                WriteMsg  PROC  NEAR
0000  B4 09                                mov    AH,9
0002  CD 21                                int     21h  ; Вызов функции
                                DOS по прерыванию
0004  C3                                ret
0005                                WriteMsg  ENDP

                                ;  Головная процедура

```

Page 1-2

```

0005          Main      PROC  FAR
0005  1E          push  DS          ;\  Сохранени
                                е адреса начала PSP в стеке
0006  2B C0      sub    AX,AX      ; > для
после□
                                ущего восстановления по
0008  50          push  AX          ;/  команде ret
                                , завершающей процедуру.
0009  B8 ---- R  mov    AX,DATA      ;
Загру□
                                зка сегментного
000C  8E D8      mov    DS,AX      ;
регио□
                                тра данных.
000E  BA 0000 R  mov    DX, OFFSET HELLO ;
Выво□
                                на экран первой
0011  E8 0000 R  call   WriteMsg      ;
стро□
                                и приветствия.
0014  BA 0010 R  mov    DX, OFFSET GREETING
; Выво□
                                на экран второй
0017  E8 0000 R  call   WriteMsg      ;
стро□
                                и приветствия.
001A  CB          ret                ;
Выхо□
                                в DOS по команде,
                                ;
нахо□
                                ящейся в 1-ом слове PSP.
001B          Main      ENDP
001B          CODE      ENDS
                                END Main

```

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	

52 Source Lines
52 Total Lines
13 Symbols

47984 + 459273 Bytes symbol space free

0 Warning Errors
0 Severe Errors