

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: «Освоение трансляции, выполнения и отладки программ
на языке Ассемблера процессора Intel X86»

Студент гр. 2384

Кузьминых Е.М.

Преподаватель

Морозов С.М.

Санкт-Петербург

2023

Цель работы.

Изучить работу программ на языке Ассемблера, сделать трансляцию, компоновку и отладку программ, зафиксировать изменение регистров во время исполнения программы.

Задание.

1. Загрузить файл *hello1.asm* из каталога \лаборат_работы в каталог \MASM.
2. Просмотреть программу в режиме редактирования, разобраться в структуре и реализации каждого сегмента программы. Непонятные фрагменты прояснить у преподавателя. Строку-приветствие преобразовать в соответствии со своими личными данными.
3. Протранслировать программу с помощью строки *> masm имя_файла.asm* с созданием объектного файла *имя_файла.obj* и файла диагностических сообщений (файла листинга) *имя_файла.lst*. Объяснить и исправить синтаксические ошибки, если они будут обнаружены транслятором и представлены в файле листинга. Повторить трансляцию программы до получения объектного файла(модуля).
4. Скомпоновать загрузочный модуль (*имя_файла.exe*) с помощью строки *> link имя_файла.obj* с созданием загрузочного модуля (*имя_файла.exe*) и файла карты памяти (*имя_файла.map*). По карте памяти оценить размещение и длину сегментов программы.
5. Выполнить программу в автоматическом режиме путем набора строки *> имя_файла.exe* и убедиться в ее работоспособности (результат выполнения посмотреть в режиме отображения экрана пользователя, получаемого набором клавиш *ctrl^O*);
6. Выполнить программу *hello1* в пошаговом режиме под управлением отладчика: *> afd имя_файла.exe* с фиксацией содержимого используемых регистров и ячеек памяти до и после выполнения команды. Обычные команды

выполняются по F1, а прерывания Int - по F2, чтобы не входить внутрь обработчика прерываний. Перемещения маркера между полями экрана управляется клавишами: F7(up), F8(down), F9(left), F10(right). В пределах поля перемещение маркера управляется клавиатурой (->, <-, и др.) Выход из отладчика - по команде *Quit*. Записать содержимое всех регистров процессора, включая сегментные, перед выполнением 1-ой команды. Результаты прогона показанном на примере одной команды в табл.1, и подписаны преподавателем.

Адрес команды	Символический код команды	16-ричный код команды	Содержимое	Рег-нов и Яч.
			До вып-я ком.	После вып-я
0003	Mov ds, Ax	8E D8	(ax)=2D87 (ds) = 2D75 (ip)=0003	(ax)=2D87 (ds) = 2D87 (ip)=0005

Табл.1

Часть 2

1. Просмотреть программу *hello2.asm* в режиме редактирования, разобраться в структуре и реализации каждого сегмента программы. Обратит внимание на использование полного описания сегментов и на использование процедуры для вывода строки. Непонятные фрагменты прояснить у преподавателя. Строки-приветствия преобразовать в соответствии со своими пожеланиями и личными данными.

2. Протранслировать программу *hello2.asm* с помощью транслятора *masm* с созданием объектного файла и файла диагностических сообщений (файла листинга). Объяснить и исправить синтаксические ошибки, если они будут обнаружены транслятором. Повторить трансляцию программы до получения объектного модуля.

3. Скомпоновать загрузочный модуль *hello2.exe* с помощью компоновщика *Link*.

4. Выполнить программу в автоматическом режиме и убедиться в ее работоспособности.

5. Выполнить программу *hello2* в пошаговом режиме под управлением отладчика *afd* с фиксацией содержимого используемых регистров и ячеек памяти до и после выполнения команд. Перед выполнением 1-ой команды записать содержимое сегментных регистров процессора.

6. Результаты прогона программы под управлением отладчика должны быть подписаны преподавателем.

Выполнение работы.

1. Содержание файла *hello1.asm*:

```
DOSSEG
.MODEL    SMALL
.STACK    100h
.DATA
Greeting LABEL BYTE
DB 'Вас приветствует ст.гр.2384 - Кузьминых Егор',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
```

2. Процесс трансляции программы, получение файлов *hello1.lst* и *hello1.obj*

```
F:\>masm.exe hello1.asm
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.

Object filename [hello1.OBJ]:
Source listing [NUL.LST]: hello1.lst
Cross-reference [NUL.CRF]:

47994 + 461313 Bytes symbol space free

0 Warning Errors
0 Severe Errors
```

3. Процесс компоновки файла и создания файла *hello1.exe*

```
F:\>link.exe hello1.obj

Microsoft (R) Overlay Linker Version 3.64
Copyright (C) Microsoft Corp 1983-1988. All rights reserved.

Run File [HELLO1.EXE]:
List File [NUL.MAP]: hello1.map
Libraries [.LIB]:
```

4. Запуск программы

```
F:\>keyb ru 866
Keyboard layout ru loaded for codepage 866

F:\>hello1.exe
Вас приветствует ст.гр.2384 – Кузьминых Е.М.
```

5. Процесс отладки программы в *afd*

```
AX 0000    SI 0000    CS 11AC    IP 0010
BX 0000    DI 0000    DS 119C
CX 004F    BP 0000    ES 119C    HS 119C
DX 0000    SP 0100    SS 11B1    FS 119C
```

IP	Машинный код инструкции	Мнемонический код инструкции	Содержимое регистров и памяти	
			До выполнения	После выполнения
0010	B8 AE 11	MOV AX,11AE	AX = 0000 IP = 0010	AX = 11AE IP = 0013
0013	8E D8	MOV DS,AX	DS = 119C	DS = 11AE

			IP = 0013	IP = 0015
0015	BA 00 00	MOV DX,0000	DX = 0000 IP = 0015	DX = 0000 IP = 0018
0018	B4 09	MOV AH,09	AX = 11AE IP = 0018	AX = 09AE IP = 001A
001A	CD 21	INT 21	IP = 001A	IP = 001C
001C	B4 4C	MOV AH,4C	AX = 09AE IP = 001C	AX = 4CAE IP = 001E
001E	CD 21	INT 21	IP = 001E	IP = 0010

Таблица 1

Часть 2

1. Содержание 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.2384 - Кузьминых Е.М.$'
DATA ENDS

CODE SEGMENT
        ASSUME CS:CODE, DS:DATA, SS:AStack

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

```

2. Процесс трансляции программы, получение файлов *hello2.lst* и *hello2.obj*

```

F:\>masm.exe hello2.asm
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.

Object filename [hello2.OBJ]:
Source listing [NUL.LST]: hello2.lst
Cross-reference [NUL.CRF]:

48002 + 461305 Bytes symbol space free

0 Warning Errors
0 Severe Errors

```

3. Процесс компоновки файла и создания файла *hello2.exe*

```

F:\>link.exe hello2.obj

Microsoft (R) Overlay Linker Version 3.64
Copyright (C) Microsoft Corp 1983-1988. All rights reserved.

Run File [HELLO2.EXE]:
List File [NUL.MAP]: hello2.map
Libraries [.LIB]:

```

4. Запуск программы

```

F:\>hello2.exe
Здравствуйте!
Вас приветствует ст.гр.2384 – Кузьминых Е.М.

```

5. Выполнение программы в *afd*

AX	0000	SI	0000	CS	11B2	IP	0005
BX	0000	DI	0000	DS	119C		
CX	007B	BP	0000	ES	119C	HS	119C
DX	0000	SP	0018	SS	11AC	FS	119C

IP	Машинный код инструкции	Мнемонический код инструкции	Содержимое регистров и памяти	
			До выполнения	После выполнения
0005	1E	PUSH DS	SP = 0018 IP = 0005	SP = 0016 IP = 0006
0006	2B C0	SUB AX,AX	AX = 0000 IP = 0006	AX = 0000 IP = 0008
0008	50	PUSH AX	SP = 0016 IP = 0008	SP = 0014 IP = 0009
0009	B8 AE 11	MOV AX,11AE	AX = 0000 IP = 0009	AX = 11AE IP = 000C
000C	8E D8	MOV DS,AX	DS = 119C IP = 000C	DS = 11AE IP = 000E
000E	BA 00 00	MOV DX,0000	DX = 0000 IP = 000E	DX = 0000 IP = 0011
0011	E8 EC FF	CALL 0000	SP = 0014 IP = 0011	SP = 0012 IP = 0000
0000	B4 09	MOV AH,09	AX = 11AE IP = 0000	AX = 09AE IP = 0002
0002	CD 21	INT 21	IP = 0002	IP = 0004
0004	C3	RET	SP = 0012	SP = 0014

			IP = 0004	IP = 0014
0014	BA 10 00	MOV DX,0010	DX = 0000 IP = 0014	DX = 0010 IP = 0017
0017	E8 E6 FF	CALL 0000	SP = 0014 IP = 0017	SP = 0012 IP = 0000
0000	B4 09	MOV AH,09	AX = 09AE IP = 0000	AX = 09AE IP = 0002
0002	CD 21	INT 21	IP = 0002	IP = 0004
0004	C3	RET	SP = 0012 IP = 0004	SP = 0014 IP = 001A
001A	CB	RET Far	SP = 0014 CS = 11B2 IP = 001A	SP = 0018 CS = 119C IP = 0000
0000	CD 20	INT 20	AX = 09AE CX = 007B CS = 119C DS = 11AE	AX = 0000 CX = 0000 CS = 11B2 DS = 119C

Таблица 2

Тестирование.

Название программы	Результат работы программы
HELLO1.EXE	'Вас приветствует ст.гр.2384 - Кузьминых Е.М.
HELLO2.EXE	Здравствуйте! Вас приветствует ст.гр.2384 - Кузьминых Е.М.

Таблица 3 — Результаты тестирования

Выводы.

Были изучены основы программирования языке Ассемблера, трансляции и отладки в эмуляторе DOSBox.

ПРИЛОЖЕНИЕ А

Файл hello1.asm

```
DOSSEG
.MODEL    SMALL
.STACK    100h
.DATA
Greeting LABEL BYTE
DB 'Вас приветствует ст.гр.2384 - Кузьминых Е.М.',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 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.2384 - Кузьминых Е.М.$'
DATA ENDS

CODE SEGMENT
ASSUME CS:CODE, DS:DATA, SS:AStack

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

```

Файл hello1.lst

Microsoft (R) Macro Assembler Version 5.10
9/15/23 16:54:54

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.2384 - Кузьминых
                                E.M.',13,10,'$'
                                A8 A2 A5 E2 E1 E2
                                A2 E3 A5 E2 20 E1
                                E2 2E A3 E0 2E 32
                                33 38 34 20 2D 20
                                8A E3 A7 EC AC A8
                                AD EB E5 20 85 2E
                                8C 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

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
Microsoft (R) Macro Assembler Version 5.10
16:54:54

9/15/23

Symbols-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	

19 Source Lines
19 Total Lines
19 Symbols

47994 + 463361 Bytes symbol space free

0 Warning Errors
0 Severe Errors

Файл hello2.lst

Microsoft (R) Macro Assembler Version 5.10

9/15/23 17:14:14
Page 1-1

= 0024 EOFLine EQU '\$'

```

0000          AStack  SEGMENT  STACK
0000 000C[          DW 12 DUP(?)
      ????
```

]

```

0018                                AStack    ENDS

0000                                DATA      SEGMENT

0000  87 A4 E0 A0 A2 E1      HELLO      DB 'Здравствуйте!', 0AH, 0DH, EOFLine
      E2 A2 E3 A9 E2 A5
      21 0A 0D 24
0010  82 A0 E1 20 AF E0      GREETING  DB 'Вас приветствует ст.гр.2384 - Куз
      бминых Е.М.$'
      A8 A2 A5 E2 E1 E2
      A2 E3 A5 E2 20 E1
      E2 2E A3 E0 2E 32
      33 38 34 20 2D 20
      8A E3 A7 EC AC A8
      AD EB E5 20 85 2E
      8C 2E 24
003D                                DATA      ENDS

0000                                CODE       SEGMENT
                                           ASSUME CS:CODE, DS:DATA, SS:AStack

0000                                WriteMsg  PROC   NEAR
0000  B4 09                                mov    AH,9
0002  CD 21                                int     21h
0004  C3                                ret
0005                                WriteMsg  ENDP

0005                                Main      PROC   FAR
0005  1E                                push   DS
0006  2B C0                                sub     AX,AX
0008  50                                push   AX
0009  B8 ---- R                          mov     AX,DATA
000C  8E D8                                mov     DS,AX
000E  BA 0000 R                          mov     DX, OFFSET HELLO
0011  E8 0000 R                          call    WriteMsg

```

```

0014  BA 0010 R          mov    DX, OFFSET GREETING
0017  E8 0000 R          call   WriteMsg
001A  CB                 ret

```

```

001B                      Main      ENDP
001B                      CODE      ENDS
                                END Main

```

Segments and Groups:

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

```

48 Source Lines
48 Total Lines
13 Symbols

```

48002 + 463353 Bytes symbol space free

```

0 Warning Errors
0 Severe Errors

```

Файл hello1.map

Start	Stop	Length	Name	Class
00000H	0001FH	00020H	_TEXT	CODE
00020H	0004EH	0002FH	_DATA	DATA
00050H	0014FH	00100H	STACK	STACK

Origin	Group
0002:0	DGROUP

Файл hello2.map

Start	Stop	Length	Name	Class
00000H	00017H	00018H	ASTACK	
00020H	0005CH	0003DH	DATA	
00060H	0007AH	0001BH	CODE	

Program entry point at 0006:0005