

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: Трансляция, отладка и выполнение
программ на языке Ассемблера

Студент гр. 0382

Осинкин Е. А.

Преподаватель

Ефремов М. А.

Санкт-Петербург

2021

Цель работы.

Изучение и работа с трансляцией, отладкой и выполнение программ на языке Ассемблера.

Задание.

Часть 1

1. Просмотреть программу hello1.asm, которая формирует и выводит на экран приветствие пользователя с помощью функции ОС MSDOS, вызываемой через прерывание с номером 21H (команда Int 21h).

Выполняемые функцией действия и задаваемые ей параметры - следующие:

- обеспечивается вывод на экран строки символов, заканчивающейся знаком "\$";
- требуется задание в регистре ah номера функции, равного 09h, а в регистре dx - смещения адреса выводимой строки;
- используется регистр ax и не сохраняется его содержимое.

2. Разобраться в структуре и реализации каждого сегмента программы. Непонятные фрагменты прояснить у преподавателя. Строку-приветствие преобразовать в соответствии со своими личными данными.

3. Загрузить файл hello1.asm из каталога Задания в каталог Masm.

4. Протранслировать программу с помощью строки

> masm hello1.asm

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.

5. Скомпоновать загрузочный модуль с помощью строки

> link hello1.obj

с созданием карты памяти и исполняемого файла hello1.exe.

6. Выполнить программу в автоматическом режиме путем набора строки

> hello1.exe

убедиться в корректности ее работы и зафиксировать результат выполнения в протоколе.

7. Запустить выполнение программы под управлением отладчика с помощью команды

> afd hello1.exe

Записать начальное содержимое сегментных регистров CS, DS, ES и SS. Выполнить программу в пошаговом режиме с фиксацией используемых регистров и ячеек памяти до и после выполнения каждой команды.

Часть 2

Выполнить пункты 1 - 7 части 1 настоящего задания применительно к программе hello2.asm, приведенной в каталоге Задания, которая выводит на экран приветствие пользователя с помощью процедуры WriteMsg, а также использует полное определение сегментов. Сравнить результаты прогона под управлением отладчика программ hello1 и hello2 и объяснить различия в размещении сегментов.

Выполнение работы.

1. Выполнение первой части:

- 1.1. Просмотрен код программы hello1.asm, прочитаны комментарии к коду, разобрано действие каждой строки.
- 1.2. Изменена строка Greeting в соответствии с личными данными.
- 1.3. Была выполнена трансляция файла hello1.asm в hello1.obj, создан файл листинга hello1.lst. Во время трансляции и создания файла не было обнаружено синтаксических ошибок.
- 1.4. Была выполнена компоновка объектного файла, создание исполняемого файла hello1.exe.
- 1.5. Файл был запущен в автоматическом режиме. Было выведено на экран: «Hello, I am from 0382 - Osinkin E.A.»
- 1.6. Исполняемый файл запущен при помощи отладчика.

Результаты представлены в таблице 1:

Таблица 1 – Результаты отладки hello1.exe

Начальные состояния регистров:

CS = 1A05, DS=19F5, ES=19F5, SS=1A0C

Адрес команд	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0010	MOV AX, 1A07	B8071A	AX = 0000 IP = 0010	AX = 1A07 IP = 0013
0013	MOV DS, AX	8ED8	DS = 19F5 IP = 0013	DS = 1A07 IP = 0015
0015	MOV DX, 0000	BA0000	DX = 0000 IP = 0015	DX = 0000 IP = 0018
0018	MOV AH, 09	B409	AX = 1A07 IP = 0018	AX = 0907 IP = 001A
001A	INT 21	CD21	IP = 001A	IP = 001A
001C	MOV AH, 4C	B44C	AX = 0907 IP = 001C	AX = 4C07 IP = 001E
001E	INT 21	CD21	IP = 001E	IP = 14A0

2. Выполнение второй части:

2.1. Также просмотрен код программы hello2.asm.

2.2. Изменена строка Greeting.

2.3. Была выполнена трансляция файла hello2.asm в hello2.obj, создан файл листинга hello2.lst. Во время трансляции и создания файла не было обнаружено синтаксических ошибок.

2.4. Была выполнена компоновка объектного файла, создание исполняемого файла hello2.exe.

2.5. Файл был запущен в автоматическом режиме. Было выведено на экран: «Hello Worlds!» «Student from 0382 - Osinkin E.A.»

2.6. Исполняемый файл запущен при помощи отладчика.

Результаты представлены в таблице 2:

Таблица 2 – Результаты отладки

hello2.exe

Начальные состояния регистров: CS = 1A0B, DS=19F5,

ES=19F5, SS=1A0C

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0005	PUSH DS	1E	IP = 0005 SP = 0018 Stack +0 0000	IP = 0006 SP = 0016 Stack +0 19F5
0006	SUB AX, AX	2BC0	AX = 0000 IP = 0006	AX = 0000 IP = 0008
0008	PUSH AX	50	IP = 0008 Stack +0 19F5 +2 0000	IP = 0009 Stack +0 0000 +2 19F5
0009	MOV AX, 1A07	B8071A	AX = 0000 IP = 0009	AX = 1A07 IP = 000C
000C	MOV DS, AX	8ED8	IP = 000C DS = 19F5	IP = 000E DS = 1A07
000E	MOV DX, 0000	BA0000	DX = 0000 IP = 000E	DX = 0000 IP = 0011
0011	CALL	E8ECFF	IP = 0011 Stack +0 0000 +2 19F5	IP = 0000 Stack +0 0014 +2 0000 +4 19F5

0000	MOV AH, 09	B409	IP = 0000 AH = 1A07	IP = 0002 AH = 0907
0002	INT 21	CD21	IP = 0002	IP = 0004
0004	RET	C3	IP = 0004 Stack +0 0014 +2 0000 +4 19F5	IP = 0014 Stack +0 0000 +2 19F5
0014	MOV DX, 0010	BA1000	IP = 0014 DX = 0000	IP = 0017 DX = 0010
0017	CALL 0000	E8E6FF	IP = 0017 Stack +0 0000 +2 19F5	IP = 0000 Stack +0 001A +2 0000 +4 19F5
0000	MOV AH, 09	B409	IP = 0000 AX = 0907	IP = 0002 AX = 0907
0002	INT 21	CD21	IP = 0002	IP = 0004
0004	RET	C3	IP = 0004 Stack +0 001A +2 0000 +4 19F5	IP = 001A Stack +0 0000 +2 19F5
001A	RET FAR	CB	IP = 001A Stack +0 0000 +2 19F5	IP = 0000 Stack +0 0000
0000	INT 20	CD20	IP = 0000	IP = 0000

Выводы.

Были изучены основы синтаксиса языка ассемблер, а также процесс трансляции, компоновки и запуска кода программы.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

hello1.asm:

```
DOSSEG
.MODEL    SMALL
.STACK    100h
.DATA
Greeting LABEL BYTE
    DB 'Hello, I am from 0382 Osinkin - E.A.',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
```

hello2.asm:

```
EOFLine EQU '$'

ASSUME CS:CODE, SS:AStack

AStack    SEGMENT    STACK
            DW 12 DUP('!')
AStack    ENDS

DATA      SEGMENT
HELLO      DB 'Hello Worlds!', 0AH, 0DH,EOFLine
GREETING DB 'Student from 0382 - Osinkin E.A.$'
DATA      ENDS

CODE      SEGMENT
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
```

ПРИЛОЖЕНИЕ Б

ЛИСТИНГИ

hello1.lst:

#Microsoft (R) Macro Assembler Version 5.10

9/16/21 12:51:20

Page 1-1

```

                                DOSSEG
                                .MODEL  SMALL
                                .STACK  100h
                                .DATA
0000          Greeting LABEL  BYTE
0000  48 65 6C 6C 6F 2C          DB 'Hello, I am from 0382 - Osinkin E.A.',13
                                ,10,'$'
                                20 49 20 61 6D 20
                                66 72 6F 6D 20 30
                                33 38 32 20 2D 20
                                4F 73 69 6E 6B 69
                                6E 20 45 2E 41 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

9/16/21 12:51:20

Symbols-1

Segments and Groups:

	N a m e	Length	Align	Combine Class
DGROUP	GROUP		
_DATA	0027	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
```

```
17 Source Lines
17 Total Lines
19 Symbols
```

47994 + 463361 Bytes symbol space free

```
0 Warning Errors
0 Severe Errors
```

hello2.lst:

#Microsoft (R) Macro Assembler Version 5.10

9/16/21 12:49:18
Page 1-1

```
= 0024                      EOFLine EQU '$'

                        ASSUME CS:CODE, SS:AStack

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

0018                      AStack      ENDS

0000                      DATA      SEGMENT
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 0382 - Osinkin E.A.$
    ,
    74 20 66 72 6F 6D
    20 30 33 38 32 20
    2D 20 4F 73 69 6E
    6B 69 6E 20 45 2E
    41 2E 24
0031                      DATA      ENDS

0000                      CODE      SEGMENT
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

```

#Microsoft (R) Macro Assembler Version 5.10

9/16/21 12:49:18
Symbols-1

Segments and Groups:

	N a m e	Length	Align	Combine Class
ASTACK		0018	PARA	STACK
CODE		001B	PARA	NONE
DATA		0031	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	

35 Source Lines
35 Total Lines
13 Symbols

47986 + 461321 Bytes symbol space free

0 Warning Errors
0 Severe Errors