

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.

Студентка гр. 1303

Королева П.А

Преподаватель

Иванов Д.В.

Санкт-Петербург

Цель работы.

Изучить простую программу на языке Ассемблера, выполнить трансляцию, линковку и отладку с отслеживанием изменений в регистрах.

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

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

с созданием карты памяти и исполняемого файла `hello1.exe`.

6. Выполнить программу в автоматическом режиме путем набора строки

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

7. Запустить выполнение программы под управлением отладчика с помощью команды

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. Обычные команды выполняются по F1 (Step), а вызовы обработчиков прерываний (Int) - по F2 (StepProc), чтобы не входить внутрь обработчика прерываний. 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.

Результаты прогона программы под управлением отладчика должны быть представлены в виде, показанном на примере одной команды в табл.1, и подписаны преподавателем.

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. Просмотрен код файла hello1.asm и hello2.asm.
2. Строка-приветствие в файле hello1 преобразована в «Вас приветствует ст.гр.1303 - Королева П.А.».

3. Файл протранслирован с помощью команды `masm hello1.asm`, получены файлы: `hello.OBJ`, `HELLO1.LST`.
4. Файл `hello.OBJ` скомпонован, получены файлы `HELLO1.EXE`, `HELLO1.MAP`.
5. `HELLO1.EXE` запущен, в терминале выведена строка «Вас приветствует ст.гр.1303 - Королева П.А.», что является корректной работой программы.
6. `HELLO1.EXE` запущен в режиме отладчика, изменения регистров и ячеек памяти зафиксированы в табл1.
7. Аналогичные шаги проделаны с файлом `hello2.asm`.

Тексты исходных файлов программ `hello1` и `hello2` представлены в приложении А.

Тексты файлов диагностических сообщений `hello1.lst` и `hello2.lst` представлены в приложении Б

Таблица 1 – результаты отладки файла HELLO1.EXE

Начальное содержимое DS, SS, ES, CS:

DS = 19F5

SS = 1A0A

ES = 19F5

CS = 1A05

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

Таблица 2 - результаты отладки файла HELLO2.EXE

Начальное содержимое DS, SS, ES, CS:

DS = 19F5

SS = 1A0A

ES = 19F5

CS = 1A0B

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 +2 0000 +4 0000 +6 0000	AX = 0000 DS = 19F5 IP = 0006 SP = 0016 Stack +0 19F5 +2 0000 +4 0000 +6 0000
0006	SUB AX, AX	2BC0	AX = 0000 DS = 19F5 IP = 0006 SP = 0016 Stack +0 19F5 +2 0000 +4 0000 +6 0000	AX = 0000 DS = 19F5 IP = 0008 SP = 0016 Stack +0 19F5 +2 0000 +4 0000 +6 0000
0008	PUSH AX	50	AX = 0000 DS = 19F5 IP = 0008 SP = 0016 Stack +0 19F5 +2 0000 +4 0000 +6 0000	AX = 0000 DS = 19F5 IP = 0009 SP = 0014 Stack +0 0000 +2 19F5 +4 0000 +6 0000
0009	MOV AX, 1A07	B8071A	AX = 0000 DS = 19F5 IP = 0009 SP = 0014 Stack +0 0000 +2 19F5 +4 0000 +6 0000	AX = 1A07 DS = 19F5 IP = 000C SP = 0014 Stack +0 0000 +2 19F5 +4 0000 +6 0000
000C	MOV DS, AX	8ED8	AX = 1A07 DS = 19F5 IP = 000C SP = 0014 Stack +0 0000 +2 19F5 +4 0000 +6 0000	AX = 1A07 DS = 1A07 IP = 000E SP = 0014 Stack +0 0000 +2 19F5 +4 0000 +6 0000
000E	MOV DX, 0000	BA0000	AX = 1A07 DS = 1A07 IP = 000E	AX = 1A07 DS = 1A07 IP = 0011

			SP = 0014 Stack +0 0000 +2 19F5 +4 0000 +6 0000	SP = 0014 Stack +0 0000 +2 19F5 +4 0000 +6 0000
0011	CALL 0000	E8ECFF	AX = 1A07 DS = 1A07 IP = 0011 SP = 0014 Stack +0 0000 +2 19F5 +4 0000 +6 0000	AX = 1A07 DS = 1A07 IP = 0000 SP = 0012 Stack +0 0014 +2 0000 +4 19F5 +6 0000
0000	MOV AH, 09	B409	AX = 1A07 DS = 1A07 IP = 0000 SP = 0012 Stack +0 0014 +2 0000 +4 19F5 +6 0000	AX = 0907 DS = 1A07 IP = 0002 SP = 0012 Stack +0 0014 +2 0000 +4 19F5 +6 0000
0002	INT 21	CD21	AX = 0907 DS = 1A07 IP = 0002 SP = 0012 Stack +0 0014 +2 0000 +4 19F5 +6 0000	AX = 0907 DS = 1A07 IP = 0004 SP = 0012 Stack +0 0014 +2 0000 +4 19F5 +6 0000
0004	RET	C3	AX = 0907 DS = 1A07 IP = 0004 SP = 0012 Stack +0 0014 +2 0000 +4 19F5 +6 0000	AX = 0907 DS = 1A07 IP = 0014 SP = 0014 Stack +0 0000 +2 19F5 +4 0000 +6 0000
0014	MOV DX, 0010	BA1000	AX = 0907 DS = 1A07 IP = 0014 SP = 0014 DX = 0000 Stack +0 0000 +2 19F5 +4 0000 +6 0000	AX = 0907 DS = 1A07 IP = 0017 SP = 0014 DX = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000

0017	CALL 0000	E8E6FF	AX = 0907 DS = 1A07 IP = 0017 SP = 0014 Stack +0 0000 +2 19F5 +4 0000 +6 0000	AX = 0907 DS = 1A07 IP = 0000 SP = 0012 Stack +0 001A +2 0000 +4 19F5 +6 0000
0000	MOV AH, 09	B409	AX = 0907 DS = 1A07 IP = 0000 SP = 0012 Stack +0 001A +2 0000 +4 19F5 +6 0000	AX = 0907 DS = 1A07 IP = 0002 SP = 0012 Stack +0 001A +2 0000 +4 19F5 +6 0000
0002	INT 21	CD21	AX = 0907 DS = 1A07 IP = 0002 SP = 0012 Stack +0 001A +2 0000 +4 19F5 +6 0000	AX = 0907 DS = 1A07 IP = 0004 SP = 0012 Stack +0 001A +2 0000 +4 19F5 +6 0000
0004	RET	C3	AX = 0907 DS = 1A07 IP = 0004 SP = 0012 Stack +0 001A +2 0000 +4 19F5 +6 0000	AX = 0907 DS = 1A07 IP = 001A SP = 0014 Stack +0 0000 +2 19F5 +4 0000 +6 0000
001A	RET Far	CB	AX = 0907 DS = 1A07 IP = 001A SP = 0014 CS = 1A0B Stack +0 0000 +2 19F5 +4 0000 +6 0000	AX = 0907 DS = 1A07 IP = 0000 SP = 0018 CS = 19F5 Stack +0 0000 +2 0000 +4 0000 +6 0000
0000	INT 20	CD20	AX = 0907 DS = 1A07 IP = 0000 SP = 0018 Stack +0 0000	AX = 0000 DS = 19F5 IP = 0005 SP = 0018 Stack +0 0000

			+2 0000	+2 0000
			+4 0000	+4 0000
			+6 0000	+6 0000

Выводы.

Была изучена простая программа на языке Ассемблер, выполнена трансляция, линковка и отладка с отслеживанием изменений в регистрах.

ПРИЛОЖЕНИЕ А

Название файла: hello1.asm

```
DOSSEG
.MODEL
.STACK
.DATA
Greeting LABEL BYTE
    DB 'Вас приветствует ст.гр.1303 - Королева П.А.',13,10,'$'
.CODE
mov ax, @data
mov ds, ax
mov dx, OFFSET Greeting

DisplayGreeting:
    mov ah, 9
    int 21h
    mov ah, 4ch
    int 21h
    ENDC
```

Название файла: 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 1303 - Koroleva Polina$'
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

Название файла: HELLO1.LST

Microsoft (R) Macro Assembler Version 5.10
25:41:40

9/26/22

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 - Королева
П
                .A.',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
                8A AE E0 AE AB A5
                A2 A0 20 8F 2E 80
                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
25:41:40

9/26/22

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

```

```

17 Source Lines
17 Total Lines
19 Symbols

```

47994 + 463361 Bytes symbol space free

```

0 Warning Errors
0 Severe Errors

```

Название файла: HELLO2.LST

Microsoft (R) Macro Assembler Version 5.10
25:41:51

9/26/22

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 1303 - Koroleva
Poli
                                na$'
    74 20 66 72 6F 6D
    20 31 33 30 33 20
    2D 20 4B 6F 72 6F
    6C 65 76 61 20 50

```

```

        6F 6C 69 6E 61 24
0034          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
 25:41:51

9/26/22

Symbols-1

Segments and Groups:

N a m e	Length	Align	Combine Class
ASTACK	0018	PARA	STACK
CODE	001B	PARA	NONE
DATA	0034	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

38 Source Lines
38 Total Lines
13 Symbols

47986 + 461321 Bytes symbol space free

0 Warning Errors
0 Severe Errors