

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

Студента гр. 9382

Иерусалимов Н.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

Цель работы.

Освоение трансляции, выполнения и отладки программ на языке Ассемблера.

Ход работы.

Файл hello1.

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:

- a. обеспечивается вывод на экран строки символов, заканчивающейся знаком "\$";
 - b. требуется задание в регистре ah номера функции, равного 09h, а в регистре dx - смещения адреса выводимой строки;
 - c. 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

a. link hello1.obj

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

a. 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. 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.

Таблица - 1

Адрес команды	Символьный код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0010	MOV AX, 1A07	B8AE11	AX = 0000 IP = 0010	AX = 1A07 IP = 0013
0013	MOV DS,AX	8ED8	AX = 1A07 DS = 19F5 IP = 0013	AX = 1A07 DS = 1A07 IP = 0015
0015	MOV DX, 0000	BA0000	IP = 0015 DX=0000	IP = 0018 DX=0000
0018	MOV AH, 09	B409	AX = 1A07 IP = 0018	AX = 0907 IP = 001A
001A	INT 21	CD21	IP = 001A	IP = 001C
001C	MOV AH,4C	B44C	AX = 0907 IP = 001C	AX = 4C07 IP = 001E
001E	INT 21	CD21	AX = 4CAE IP = 001E	AX = 0000 IP = 0010

Hello1.lst:

Microsoft (R) Macro Assembler Version 5.10

11/14/20 13:58:0

Page 1-1

```

DOSSEG
; Директива DOSSEG приводит к тому, что сегмент
; в программе

;Ассемблера будут сгруппированы в соответствии
и с соглашениями по

;упорядочиванию сегментов фирмы Microsoft.
.MODEL SMALL
; позволяет указывать используемую модель памят
и и соглашение о вызовах.
.STACK 100h
; Установка размера стека
.DATA
; мы резервируем память под данные и сразу же и
нициализируем их, т.е. задаём им начальные знач
ения
0000 Greeting LABEL BYTE

0000 59 6F 24 DB 'Yo','$'
.CODE ; отмеч
ает начало сегмента кода
0000 B8 ---- R mov ax, @data
??? ? DS х??? Ё ? —
0003 8E D8 mov ds, ax
?Ё? х ЁЁ??
0005 BA 0000 R mov dx, OFFSET Greeting

0008 DisplayGreeting:
0008 B4 09 mov ah, 9 ; для
вывода
000A CD 21 int 21h ;
000C B4 4C mov ah, 4ch ; для
завершения программы
000E CD 21 int 21h ;
```

END

__Microsoft (R) Macro Assembler Version 5.10

11/14/20 13:58:0

Symbols-1

Segments and Groups:

	N a m e	Length	Align	Combine	Class
DGROUP				GROUP
__DATA	0003	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	HELL01	
@VERSION	TEXT	510	

19 Source Lines
 19 Total Lines
 19 Symbols

47994 + 461313 Bytes symbol space free

0 Warning Errors
 0 Severe Errors

Файл hello2.

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.

Таблица - 2

Адрес команды	Символьный код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения

0005	PUSH DS	1E	IP 0005 Stack +0 0000 DS=19F5 SP=0018	IP 0006 Stack +0 19F5 DS=19F5 SP=0016
0006	SUB AX,AX	2BC0	IP = 0006 AX=0000 SP=0016	IP = 0008 AX=0000 SP=0016
0008	PUSH AX	50	Stack +0 19F5 +2 0000 IP = 0008 AX=0000	Stack +0 0000 +0 19F5 IP = 0009 AX=0000
0009	MOV AX, 1A07	B8071A	AX = 0000 IP=0009	AX = 1A07 IP=000C
000C	MOV DS,AX	8ED8	DS = 19F5 IP = 000C	DS = 1A07 IP = 000E
000E	MOV DX, 0000	BA0000	IP = 000E DX=0000	IP = 0011 DX=0000
0011	CALL 0000	E8ECFF	+0 0000 +0 19F5 +4 0000 IP = 0011 SP=0014	+0 0014 +2 0000 +4 19F5 IP = 0000 SP=0012
0000	MOV AH 09	B409	+ 0 0014 +2 0000 +4 19F5 AX = 1A07 IP = 0000 SP=0012	+0 0014 +2 0000 +4 19F5 AX = 0907 IP = 0002 SP=0012
0002	INT 21	CD21	IP = 0002 AX = 0907	IP = 0004 AX = 0907

0004	RET	C3	+0 0014 +2 0000 +4 19F5 IP = 0004 SP=0012	+0 0000 +2 19F5 +4 0000 IP = 0014 SP=0014
0014	MOV DX 0010	BA1000	+ 0 0000 +2 19F5 DX = 0000 IP = 0014	+ 0 0000 +2 119C DX = 0010 IP = 0017
0017	CALL 0000	E8E6FF	+0 0000 +2 19F5 +4 0000 IP = 0017 SP=0014	+0 001A +2 0000 +4 19F5 IP = 0000 SP=0012
0000	MOV AH 09	B409	IP = 0000 AX=0907	IP = 0002 AX=0907
0002	INT 21	CD 21	IP = 0002	IP = 0004
0004	RET	C3	+0 001A +2 0000 +4 19F5 IP = 0004 SP=0012	+0 0000 +2 19F5 +4 0000 IP = 001A SP=0014
001A	RET FAR	CB	+0 0000 +2 19F5 +4 0000 SP=0014 IP = 001A	+0 0000 +2 0000 +4 0000 SP=0018 IP = 0000
0000	INT 20	CD20	IP = 0000 DS = 1A07 AX = 0907 CS = 19F5	IP = 0005 DS = 19F5 AX = 0000 CS = 1A0B

Hello2.lst:


```

0008 50                push AX          ;/ €- -г ret, $ ўҒа
и ойҒ@ ĩа®жҒҒаг.
0009 B8 ---- R        mov AX,DATA      ; ‡ Јаг$€
бҒЈ-Ғ-в-®Ј®
000C 8E D8            mov DS,AX        ; аҒЈЁбва
Ғ --ле.
000E BA 0000 R        mov DX, OFFSET HELLO ; ,лў®Ғ -
н€а - ĩҒаў®@
0011 E8 0000 R        call WriteMsg     ; бва®€Ё ĩа
ЁўҒвбвўЁп.
0014 BA 0010 R        mov DX, OFFSET GREETING ; ,лў®Ғ -
н€а - ўв®а®@
0017 E8 0000 R        call WriteMsg     ; бва®€Ё ĩа
ЁўҒвбвўЁп.
001A CB              ret              ; ,ле®Ғ ў D
OS ĩ® €- -г,
; - е®ҒпйҒ@

бп ў 1-®- б«®ўҒ PSP.
001B Main            ENDP
001B CODE            ENDS
END Main
__Microsoft (R) Macro Assembler Version 5.10
11/14/20 13:57:2
Symbols-1

```

Segments and Groups:

N a m e	Length	Align	Combine	Class
ASTACK	0018	PARA	STACK	
CODE	001B	PARA	NONE	;27
DATA	0033	PARA	NONE	;51

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	HELL02	
@VERSION	TEXT	510	

```

51 Source Lines
51 Total Lines
13 Symbols

```

48002 + 461305 Bytes symbol space free

```

1 Warning Errors
0 Severe Errors

```

Выводы.

В ходе выполнения данной работы был освоен навык работы с DOSBox.
Освоили трансляцию, выполнение и отладку программ на языке Ассемблера.