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

Студентка гр. 0383 _____ Александрович В.П.

Преподаватель _____ Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Познакомиться с трансляцией, отладкой и выполнением программ на языке Ассемблер.

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.

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

Адрес команды	Символический код команды	16-ичный код команды	Содержимое регистров и ячеек памяти	
			до выполнения	после выполнения
0003	Mov DS, AX	8E D8	(AX) = 2D87 (DS) = 2D75 (IP) = 0003	(AX) = 2D87 (DS) = 2D87 (IP) = 0005

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.

Часть 1.

1. Просмотрено содержание файла исходного кода hello1.asm.
2. Программа протранслирована с помощью функции masm hello1.asm, создан объектный файл hello1.obj и файл диагностических ошибок hello1.lst. Синтаксические ошибки не были обнаружены.
3. Загрузочный файл скомпонован с помощью команды link hello1.obj с созданием карты памяти и исполняемого файла hello1.exe.
4. Запущен файл hello1.exe, выведено сообщение «Вас приветствует ст. гр. 7303 – Иванов И. И.» (Для корректного вывода кириллицы была подана команда keyb ru 866).
5. Программа запущена под управлением отладчика с помощью команды afdpro.exe hello1.exe.

Начальное содержимое сегментов:

(CS) = 1A05

(DS) = 19F5

(ES) = 19F5

(SS) = 1A0A

Адрес команды	Символический код команды	16-ичный 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	(AX) = 0907

			(DS) = 1A07 (IP) = 0018	(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. Просмотрено содержание файла исходного кода hello2.asm.
2. Программа протранслирована с помощью функции masm hello2.asm, создан объектный файл hello2.obj и файл диагностических ошибок hello2.lst. Синтаксические ошибки не были обнаружены.
3. Загрузочный файл скомпонован с помощью команды link hello2.obj с созданием карты памяти и исполняемого файла hello2.exe.
4. Запущен файл hello2.exe, выведено сообщение «Hello Worlds! Student from 4350 –»
5. Программа запущена под управлением отладчика с помощью команды afdpro.exe hello2.exe.

Начальное содержимое сегментов:

(CS) = 1A0A

(DS) = 19F5

(ES) = 19F5

(SS) = 1A05

Адрес команды	Символический код команды	16-ичный 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 (SP) = 0016 Stack +0 19F5	(AX) = 0000 (DS) = 19F5 (IP) = 0008 (SP) = 0016 Stack +0 19F5
0008	Push AX	50	(AX) = 0000 (DS) = 19F5 (IP) = 0008 (SP) = 0016 Stack +0 19F5 +2 0000	(AX) = 0000 (DS) = 19F5 (IP) = 0009 (SP) = 0014 Stack +0 0000 +2 19F5
0009	Mov AX, 1A07	B8071A	(AX) = 0000 (DS) = 19F5 (IP) = 0008 (SP) = 0014 Stack +0 0000 +2 19F5	(AX) = 1A07 (DS) = 19F5 (IP) = 000C (SP) = 0014 Stack +0 0000 +2 19F5
000C	Mov DS, AX	8ED8	(AX) = 1A07 (DS) = 19F5 (IP) = 000C (SP) = 0014 Stack +0 0000 +2 19F5	(AX) = 1A07 (DS) = 1A07 (IP) = 000E (SP) = 0014 Stack +0 0000 +2 19F5
000E	Mov DX, 0000	BA0000	(AX) = 1A07 (DS) = 1A07 (IP) = 000E	(AX) = 1A07 (DS) = 1A07 (IP) = 0011

			(SP) = 0014 Stack +0 0000 +2 19F5	(SP) = 0014 Stack +0 0000 +2 19F5
0011	Call 0000	E8ECFF	(AX) = 1A07 (DS) = 1A07 (IP) = 0011 (SP) = 0014 Stack +0 0000 +2 19F5 +4 0000	(AX) = 1A07 (DS) = 1A07 (IP) = 0000 (SP) = 0012 Stack +0 0014 +2 0000 +4 19F5
0000	Mov AH, 09	B409	(AX) = 1A07 (DS) = 1A07 (IP) = 0000 (SP) = 0012 Stack +0 0014 +2 0000 +4 19F5	(AX) = 0907 (DS) = 1A07 (IP) = 0002 (SP) = 0012 Stack +0 0014 +2 0000 +4 19F5
0002	Int 21	SD21	(AX) = 0907 (DS) = 1A07 (IP) = 0002 (SP) = 0012 Stack +0 0014 +2 0000 +4 19F5	(AX) = 0907 (DS) = 1A07 (IP) = 0004 (SP) = 0012 Stack +0 0014 +2 0000 +4 19F5
0004	Ret	C3	(AX) = 0907 (DS) = 1A07 (IP) = 0004 (SP) = 0012 Stack +0 0014 +2 0000 +4 19F5	(AX) = 0907 (DS) = 1A07 (IP) = 0014 (SP) = 0014 Stack +0 0014 +2 19F5 +4 0000
0014	Mov DX, 0010	BA1000	(AX) = 0907 (DS) = 1A07 (IP) = 0014	(AX) = 0907 (DS) = 1A07 (IP) = 0017

			(SP) = 0014 Stack +0 0014 +2 19F5 +4 0000	(SP) = 0014 Stack +0 0014 +2 19F5 +4 0000
0017	Call 0000	E8ECFF	(AX) = 0907 (DS) = 1A07 (IP) = 0017 (SP) = 0014 Stack +0 0014 +2 19F5 +4 0000	(AX) = 0907 (DS) = 1A07 (IP) = 0000 (SP) = 0012 Stack +0 001A +2 0000 +4 19F5
0000	Mov AH, 09	B409	(AX) = 0907 (DS) = 1A07 (IP) = 0000 (SP) = 0012 Stack +0 001A +2 0000 +4 19F5	(AX) = 0907 (DS) = 1A07 (IP) = 0002 (SP) = 0012 Stack +0 001A +2 0000 +4 19F5
0002	Int 21	CD21	(AX) = 0907 (DS) = 1A07 (IP) = 0002 (SP) = 0012 Stack +0 001A +2 0000 +4 19F5	(AX) = 0907 (DS) = 1A07 (IP) = 0004 (SP) = 0012 Stack +0 001A +2 0000 +4 19F5
0004	Ret	C3	(AX) = 0907 (DS) = 1A07 (IP) = 0004 (SP) = 0012 Stack +0 001A +2 0000 +4 19F5	(AX) = 0907 (DS) = 1A07 (IP) = 001A (SP) = 0014 Stack +0 0000 +2 19F5 +4 0000
001A	Ret Far	CB	(AX) = 0907 (DS) = 1A07	(AX) = 0907 (DS) = 1A07

			(IP) = 001A (SP) = 0014 Stack +0 0000 +2 19F5 +4 0000	(IP) = 0000 (SP) = 0018 Stack +0 0000 +2 0000 +4 0000
0000	Int 20	CD20	(AX) = 0907 (DS) = 1A07 (IP) = 0000 (SP) = 0018 (CX) = 006B (CS) = 19F5 Stack +0 0000 +2 0000 +4 0000	(AX) = 0000 (DS) = 1A07 (IP) = 0005 (SP) = 0018 (CX) = 0000 (CS) = 1A0A Stack +0 0000 +2 0000 +4 0000

Выводы.

В ходе выполнения данной лабораторной работы был изучен процесс трансляции, отладки и выполнения программ на языке Ассемблер. Были составлены таблицы состояния программы для каждого шага.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: hello1.asm

```
DOSSEG
   .MODEL SMALL
   .STACK 100h
   .DATA
Greeting LABEL BYTE
   DB ', б ЇаЁўҐвбвўгҐв бв.Ja.7303 - €ў ®ў €.€.',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
```

Название файла: hello2.asm

```
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 4350 - $'
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
```