

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ по лабораторной работе №1
по дисциплине «Программирование»
ТЕМА: ТРАНСЛЯЦИИ, ОТЛАДКА И ВЫПОЛНЕНИЕ ПРОГРАММ НА ЯЗЫКЕ
АССЕМБЛЕРА.

Студент гр. 1303

Дауди И.Я.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

Изучить основные принципы трансляции, отладки и выполнения программ на языке Ассемблера.

Задание.

1. Просмотреть программу hello1.asm.

2. Разобраться в структуре и реализации каждого сегмента программы.

Строку-приветствие преобразовать в соответствии со своими личными данными.

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

4. Протранслировать программу 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 созданием карты памяти и исполняемого файла hello1.exe.

6. Выполнить программу в автоматическом режиме, убедиться в корректности ее работы и зафиксировать результат выполнения в протоколе.

7. Запустить выполнение программы под управлением отладчика.

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.

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:

- DS = 19F5
- SS = 1A0A
- ES = 19F5
- CS = 1A05

hello1.exe:

Адрес Команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			до выполнения	после выпол- нения
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

- DS = 19F5
- SS = 1A0A
- ES = 19F5
- CS = 1A0A

hello2.exe

Адрес Команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			до выполнения	после выполне- ния
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	(AX) = 0000 (DS) = 19F5 (IP) = 0008
0008	Push AX	50	(AX) = 0000 (DS) = 19F5 (IP) = 0008 (SP) = 0016 Stack +0 =19F5 Stack +2 =0000	(AX) = 0000 (DS) = 19F5 (IP) = 0009 (SP) = 0014 Stack+0=0000 Stack+2=19F5
0009	Mov AX, 1A07	B8071A	(AX) = 0000 (DS) = 19F5 (IP) = 0009	(AX) = 1A07 (DS) = 19F5 (IP) = 000C
000C	Mov DS, AX	8ED8	(AX) = 1A07 (DS) = 19F5 (IP) = 000C	(AX) = 1A07 (DS) = 1A07 (IP) = 000E

000E	Mov DX, 0000	BA0000	(AX) = 1A07 (DS) = 1A07 (IP) = 000E	(AX) = 1A07 (DS) = 1A07 (IP) = 0011
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	(AX) = 0907 (DS) = 1A07 (IP) = 0002
0002	Int 21	CD21	(AX) = 0907 (DS) = 1A07 (IP) = 0002	(AX) = 0907 (DS) = 1A07 (IP) = 0004
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=0000 +2=19F5 +4=0000
0014	Mov DX, 0010	BA1000	(AX) = 0907 (DS) = 1A07 (IP) = 0014 (DX) = 0000	(AX) = 0907 (DS) = 1A07 (IP) = 0017 (DX) = 0010
0017	Call 0000	E8E6FF	(AX) = 0907 (DS) = 1A07 (IP) = 0017 (SP) = 0014 Stack +0=0000 +2=19F5	(AX) = 0907 (DS) = 1A07 (IP) = 0000 (SP) = 0012 Stack+0=001A

			+4=0000	+2=0000 +4=19F5
0000	Mov AH, 09	B409	(AX) = 0907 (DS) = 1A07 (IP) = 0000	(AX) = 0907 (DS) = 1A07 (IP) = 0002
0002	Int 21	CD21	(AX) = 0907 (DS) = 1A07 (IP) = 0002	(AX) = 0907 (DS) = 1A07 (IP) = 0004
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 (IP) = 001A (CS) = 1A0B (SP) = 0014 Stack +0=0000 +2=19F5 +4=0000	(AX) = 0907 (DS) = 1A07 (IP) = 0000 (CS) = 19F5 (SP) = 0018 Stack+0=0000 +2=0000 +4=0000
0000	Int 20	CD20	(AX) = 0907 (DS) = 1A07 (IP) = 0000	(AX) = 0000 (DS) = 19F5 (IP) = 0005

Выводы

Были изучены основные принципы трансляции, отладки и выполнения программ на языке Ассемблера.