

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

Студент гр. 9383

Преподаватель

Поплавский И.

Ефремов М.А.

Санкт-Петербург

2020

Цель работы.

Освоение трансляции, выполнения и отладки программ на языке Ассемблера.

Ход работы.

Часть 1. Файл hello1.

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.

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

2. 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.

1. Скомпоновать загрузочный модуль с помощью строки

- a. link hello1.obj

2. созданием карты памяти и исполняемого файла hello1.exe.

1. Выполнить программу в автоматическом режиме путем набора строки

> hello1.exe

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

1. Запустить выполнение программы под управлением отладчика с помощью команды

а. 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	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	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 = 4C07 IP = 001E	AX = 0000 IP = 0010

Часть 2. Файл hello20.

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	CD21	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 = 1A0A

Выводы.

В результате выполнения данной работы был получен навык создания программ на языке Ассемблер. Получены основы теоретических и практических знаний о работе регистров, процессе создания программ, а также их структуре.

