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 **А**ССЕМБЛЕРА.

Студент гр.0382	Диденко Д.В.
Преподаватель	Ефремов М.А.

Санкт-Петербург 2021

Цель работы.

Научиться выполнять трансляцию, отладку и запускать программы на языке ассемблер.

Задание.

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.

Основные теоретические положения.

Лабораторная работа 1 использует 2 готовых программы на ассемблере: hello1 — составлена с использованием сокращенного описания сегментов и hello2 — составлена с полным описанием сегментов и выводом строки, оформленным как процедура. Выполнение работы состоит из двух частей.

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 ах и не сохраняется его содержимое.
 - 2. Разобраться в структуре и реализации каждого сегмента программы.
 - 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
 - > afdpro hello1.exe

Выполнение работы.

Результаты хода программы hello1.exe представлены в табл. 1.

Таблица 1 – Ход программы hello1.exe.

				Содержимое регистров и ячеек			
№	A was a state of the state of t	Символический	16-ричный код	пам	ИТК		
п/п	Адрес команды	код команды	команды	До	После		
				выполнения	выполнения		
				(AX) = 0000	(AX) = 1A07		
				(CS) = 1A05	(CS) = 1A05		
1.	0010	mov ax,1A07	B8071A	(DS) = 19F5	(DS) = 19F5		
				(ES) = 19F5	(ES) = 19F5		
				(SS) = 1A09	(SS) = 1A09		
				(AX) = 1A07	(AX) = 1A07		
				(CS) = 1A05	(CS) = 1A05		
2.	0013	Mov ds, ax	8ED8	(DS) = 19F5	(DS) = 1A07		
				(ES) = 19F5	(ES) = 19F5		
				(SS) = 1A09	(SS) = 1A09		
3.	0015	Mov dx,0000	BA0000	(AX) = 1A07	(AX) = 1A07		

				(CS) = 1A05	(CS) = 1A05
				(DS) = 1A07	(DS) = 1A07
				(ES) = 19F5	(ES) = 19F5
				(SS) = 1A09	(SS) = 1A09
				(AX) = 1A07	(AX) = 0907
				(CS) = 1A05	(CS) = 1A05
4.	0018	Mov ah,09	B409	(DS) = 1A07	(DS) = 1A07
				(ES) = 19F5	(ES) = 19F5
				(SS) = 1A09	(SS) = 1A09
	001A			(AX) = 0907	(AX) = 0907
				(CS) = 1A05	(CS) = 1A05
5.		Int 21h	CD21	(DS) = 1A07	(DS) = 1A07
				(ES) = 19F5	(ES) = 19F5
				(SS) = 1A09	(SS) = 1A09
				(AX) = 0907	(AX) = 4C07
				(CS) = 1A05	(CS) = 1A05
6.	001C	Mov ah,4ch	B44C	(DS) = 1A07	(DS) = 1A07
				(ES) = 19F5	(ES) = 19F5
				(SS) = 1A09	(SS) = 1A09
				(AX) = 4C07	
				(CS) = 1A05	
7.	001E	Int 21	CD21	(DS) = 1A07	End
				(ES) = 19F5	
				(SS) = 1A09	
i		i e	i e		

Результат работы программы hello1.exe.

```
47994 + 463361 Bytes symbol space free

0 Warning Errors
0 Severe Errors

C:\>hello1.exe
Hi Im student Danil Didenko

C:\>
```

Результаты хода программы hello2.exe представлены в табл. 2.

Таблица 2 – Ход программы hello2.exe.

№ п/п Символический код команды 16-ричный код команды Содержимое регистров и ячеек памяти До выполнения После выполнения Выполнения Выполнения (AX) = 0000 (CS) = 1 A0B (DS) = 19F5 (ES) = 19F5	1 403111	— Под пре	pawwibi nenoz					
Адрес команды								
1. 1. 1. 1. 1. 1. 1. 1.	$N_{\underline{0}}$	A	Символический	16-ричный код	пам	ИТК		
$(AX) = 0000 \\ (CS) = 1A0B \\ (DS) = 19F5 \\ (ES) = 19F5 \\ (ES) = 19F5 \\ (SS) = 1A05 \\ (SP) = 0000 \\ Stack: \\ +0:0000 \\ +4:0000 \\ +4:0000 \\ +6:0000 \\ (CS) = 1A0B \\ (DS) = 19F5 \\ (SS) = 1A05 \\ (SP) = 0016 \\ Stack: \\ +0:0000 \\ +2:0000 \\ +4:0000 \\ +6:0000 \\ (CS) = 1A0B \\ (DS) = 19F5 \\ (SS) = 1A05 \\ (SP) = 0016 \\ Stack: \\ +0:0000 \\ (CS) = 1A0B \\ (DS) = 19F5 \\ (ES) = 1405 \\ (SP) = 0016 \\ Stack: \\ +0:19F5 \\ +2:0000 \\ +4:0000 \\ +4:0000 \\ +4:0000 \\ +4:0000 \\ +6:0000 \\ (CS) = 1A0B \\ (DS) = 19F5 \\ (ES) = 1000 \\ $	Π/Π	Адрес команды	код команды	команды	До	После		
(CS) = 1A0B (DS) = 19F5 (ES) = 19F5 (ES) = 19F5 (ES) = 19F5 (ES) = 1000 (SS) = 1A05 (SP) = 0000 Stack: +0: 0000 +4: 0000 +4: 0000 +6: 0000 +6: 0000 (CS) = 1A0B (CS) = 1A05 (SP) = 0016 (CS) = 1A0B (выполнения	выполнения		
$(CS) = 1A0B \\ (DS) = 19F5 \\ (ES) = 19F5 \\ (ES) = 19F5 \\ (ES) = 1405 \\ (SS) = 1A05 \\ (SS) = 1A05 \\ (SP) = 0000 \\ Stack: \\ +0: 0000 \\ +4: 0000 \\ +4: 0000 \\ +4: 0000 \\ +6: 0000 \\ +6: 0000 \\ +6: 0000 \\ +6: 0000 \\ (CS) = 1A0B \\ (DS) = 19F5 \\ (ES) = 1000 \\ (ES) = 10000 \\ (ES) = 1000 \\ $					(AX) = 0000	(AX) = 0000		
$\begin{array}{cccccccccccccccccccccccccccccccccccc$					(CS) = 1A0B	, , ,		
1. 0005 Push DS 1E (SS) = 1A05 (SS) = 1A05 (SP) = 0000 (STack: +0: 0000 +0: 19F5 +2: 0000 +4: 0000 +6: 0000 +6: 0000 +6: 0000 (CS) = 1A0B (DS) = 19F5 (ES) = 10F5					(DS) = 19F5			
1. 0005 Push DS 1E (SP) = 0000 (SS) = 1A03 (SP) = 0016 Stack: +0: 0000 +0: 19F5 +2: 0000 +2: 0000 +4: 0000 +6: 0000 +6: 0000 +6: 0000 (CS) = 1A0B (DS) = 19F5 (ES) = 19F5 (ES) = 19F5 (ES) = 19F5 (ES) = 1405 (SR) = 0000 +2: 0000 +2: 0000 +2: 0000 (CS) = 1A0B (DS) = 19F5 (ES) = 19F5 (ES) = 19F5 (ES) = 19F5 (ES) = 19F5 +0: 19F5 +0: 19F5 +0: 19F5 +0: 19F5 +2: 0000 +2: 0000 +4: 0000 +4: 0000 +4: 0000 +6: 0000 +6: 0000 (CS) = 1A0B (DS) = 19F5 (DS) = 1405 (SS) = 140					(ES) = 19F5	(ES) = 19F5		
Stack: Stack: Stack: +0: 0000 +0: 19F5 +2: 0000 +4: 0000 +6: 0000 +6: 0000 (AX) = 0000 (CS) = 1A0B (CS) = 19F5 (ES) = 19F5 (ES) = 19F5 (ES) = 19F5 (ES) = 19F5 +0: 19F5 +2: 0000 +4: 0000 +4: 0000 +4: 0000 +4: 0000 +4: 0000 +4: 0000 +4: 0000 +6: 0000 +6: 0000 +6: 0000 +6: 0000 +6: 0000 (CS) = 1A0B (CS) = 1A05 (SS) = 0014					(SS) = 1A05	(SS) = 1A05		
2. 0006 Sub ax,ax 2BCO (SP) = 0016 Stack: +0: 19F5 +0: 19F5 +0: 19F5 +0: 2: 0000 +4: 0000 +6:	1.	0005	Push DS	1E	(SP) = 0000	(SP) = 0016		
2. 0006 Sub ax,ax 2BCO (SP) = 0016 (SP) = 0016 (ST) = 14,0000 (AX) = 0000 (AX) = 0000 (CS) = 140B (DS) = 19F5 (ES) = 19F5 (ES) = 19F5 (ES) = 19F5 (SS) = 1405 (SP) = 0016 (SP) = 19F5 (SP) = 10016 (SP) = 0014 (SP) = 0014					Stack:	Stack:		
2. 0006 Sub ax,ax 2BCO (SP) = 0016 Stack: +0: 19F5 +0: 19F5 +2: 0000 +4: 0000 +6: 0000 (CS) = 1A0B (CS) = 1A0B (SP) = 0016 Stack: +0: 19F5 +2: 0000 +2: 0000 (CS) = 1A0B (CS) = 1A0B (SP) = 0016 Stack: Stack: +0: 19F5 +0: 19F5 +2: 0000 (CS) = 1A0B (CS) = 19F5 (SS) = 1A05 (SS) = 0016 (SP) = 0014					+0: 0000	+0: 19F5		
+6: 0000 +6: 0000 (AX) = 0000 (CS) = 1A0B (CS) = 1A0B (DS) = 19F5 (DS) = 19F5 (ES) = 19F5 (ES) = 19F5 (SS) = 1A05 (SP) = 0016 Stack: Stack: +0: 19F5 +2: 0000 +4: 0000 +4: 0000 +6: 0000 (CS) = 1A0B (CS) = 1A0B (CS) = 1A05 (SP) = 0016 Stack: Stack: 0008 Push ax 50 (ES) = 19F5 (ES) = 1A0B (CS) = 1A0B (+2: 0000	+2: 0000		
$(AX) = 0000 \qquad (AX) = 0000 \qquad (CS) = 1A0B \qquad (CS) = 1A0B \qquad (DS) = 19F5 \qquad (DS) = 19F5 \qquad (ES) = 19F5 \qquad (ES) = 19F5 \qquad (ES) = 19F5 \qquad (ES) = 1405 \qquad (SS) = 1A05 \qquad (SS) = 14000 \qquad (SS) = 14000 \qquad (SS) = 14000 \qquad (SS) = 14000 \qquad (SS) = 140000 \qquad (SS) = 140000 \qquad (SS) = 140000 \qquad (SS) = 1400000 \qquad (SS) = 1400000000000000000000000000000000000$					+4: 0000	+4: 0000		
$(CS) = 1A0B \qquad (CS) = 1A0B \qquad (CS) = 1A0B \qquad (DS) = 19F5 \qquad (DS) = 19F5 \qquad (DS) = 19F5 \qquad (ES) = 19F5 \qquad (ES) = 19F5 \qquad (ES) = 19F5 \qquad (ES) = 1405 \qquad (SS) = 1A05 \qquad (SP) = 0016 \qquad Stack: \qquad +0: 19F5 \qquad +0: 19F5 \qquad +0: 19F5 \qquad +2: 0000 \qquad +2: 0000 \qquad +4: 0000 \qquad +4: 0000 \qquad +6: 0000 \qquad (CS) = 1A0B \qquad (CS) = 1A0B \qquad (CS) = 1A0B \qquad (DS) = 19F5 \qquad (DS) = 19F5 \qquad (ES) = 19F5 \qquad (ES) = 19F5 \qquad (SS) = 1A05 \qquad (SP) = 0016 \qquad (SP) = 0014 \qquad ($					+6: 0000	+6: 0000		
2. 0006 Sub ax,ax 2BCO $ (DS) = 19F5 \qquad (DS) = 19F5 \\ (ES) = 19F5 \qquad (ES) = 19F5 \\ (SS) = 1A05 \qquad (SS) = 1A05 \\ (SP) = 0016 \qquad (SP) = 0016 \\ Stack: \qquad +0: 19F5 \qquad +0: 19F5 \\ +2: 0000 \qquad +2: 0000 \\ +4: 0000 \qquad +4: 0000 \\ +6: 0000 \qquad +6: 0000 \\ (CS) = 1A0B \qquad (CS) = 1A0B \\ (DS) = 19F5 \qquad (DS) = 19F5 \\ (SS) = 1A05 \qquad (SP) = 0014 \\ (SP) = 0016 \qquad (SP) = 0014 $				2BCO	(AX) = 0000	(AX) = 0000		
(ES) = 19F5 (ES) = 19F5 (SS) = 1A05 (SS) = 1A05 (SP) = 0016 (SP) = 0016 Stack: Stack: +0: 19F5 +0: 19F5 +2: 0000 +2: 0000 +4: 0000 +6: 0000 (CS) = 1A0B (CS) = 1A0B (DS) = 19F5 (DS) = 19F5 (SS) = 1A05 (SP) = 0016 Stack: Stack: (AX) = 0000 (AX) = 0000 (CS) = 1A0B (CS) = 1A0B (CS) = 19F5 (SS) = 19F5 (SS) = 19F5 (SS) = 1A05 (SP) = 0016 (SP) = 0014					(CS) = 1A0B	(CS) = 1A0B		
2. 0006 Sub ax,ax 2BCO $ (SS) = 1A05 (SS) = 1A05 (SP) = 0016 (SP) = 0016 (SP) = 0016 (SE) = 19F5 +0: 19F5 +2: 0000 +4: 0000 +6: 0000 +6: 0000 (CS) = 1A0B (DS) = 19F5 (DS) = 19F5 (SS) = 1A05 (SP) = 0014 (SP) = 0014 (SP) = 0016 (SP) = 0016 (SP) = 0014 (SP) = 0016 (SP) = 0016 (SP) = 0014 (SP) = 0016 (SP) = 0016 (SP) = 0014 (SP) = 0016 (SP) = 0014 (SP) = 0016 (SP) = $					(DS) = 19F5	(DS) = 19F5		
2. 0006 Sub ax,ax 2BCO (SP) = 0016 Stack: (SP) = 0016 Stack: +0: 19F5 +0: 19F5 +2: 0000 +2: 0000 +4: 0000 +6: 0000 (CS) = 1A0B (CS) = 1A0B (DS) = 19F5 (ES) = 19F5 (SS) = 1A05 (SP) = 0014 (SP) = 0014					(ES) = 19F5	(ES) = 19F5		
Stack: Stack: +0: 19F5 +0: 19F5 +2: 0000 +2: 0000 +4: 0000 +6: 0000 (AX) = 0000 (AX) = 0000 (CS) = 1A0B (CS) = 1A0B (DS) = 19F5 (DS) = 19F5 (SS) = 1A05 (SP) = 0014					(SS) = 1A05	(SS) = 1A05		
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	2.	0006	Sub ax,ax					
					+0: 19F5	+0: 19F5		
+6:0000 +6:0000 $(AX) = 0000 (AX) = 0000$ $(CS) = 1A0B (CS) = 1A0B$ $(DS) = 19F5 (DS) = 19F5$ $(ES) = 19F5 (ES) = 19F5$ $(SS) = 1A05 (SS) = 1A05$ $(SP) = 0016 (SP) = 0014$					+2: 0000	+2: 0000		
3. $(AX) = 0000$ $(AX) = 0000$ $(CS) = 1A0B$ $(CS) = 1A0B$ $(DS) = 19F5$ $(DS) = 19F5$ $(ES) = 19F5$ $(SS) = 1A05$ $(SS) = 1A05$ $(SP) = 0014$					+4: 0000	+4: 0000		
3. Push ax $(CS) = 1A0B \qquad (CS) = 1A0B$ $(DS) = 19F5 \qquad (DS) = 19F5$ $(ES) = 19F5 \qquad (ES) = 19F5$ $(SS) = 1A05 \qquad (SS) = 1A05$ $(SP) = 0016 \qquad (SP) = 0014$					+6: 0000	+6: 0000		
3. Push ax $(DS) = 19F5 \qquad (DS) = 19F5$ $(ES) = 19F5 \qquad (ES) = 19F5$ $(SS) = 1A05 \qquad (SS) = 1A05$ $(SP) = 0014$					(AX) = 0000	(AX) = 0000		
3. Push ax $(ES) = 19F5 \qquad (ES) = 19F5$ $(SS) = 1A05 \qquad (SP) = 0014$					(CS) = 1A0B	(CS) = 1A0B		
(SS) = 1A05 (SP) = 0016 (SP) = 0014					(DS) = 19F5	(DS) = 19F5		
(SP) = 0016 $(SP) = 0014$	3.	0008	Push ax	50	(ES) = 19F5	(ES) = 19F5		
					(SS) = 1A05	(SS) = 1A05		
Stack: Stack:						(SP) = 0014		
, , , , , , , , , , , , , , , , , , , ,					Stack:	Stack:		

				10 1057	10.0000
				+0: 19F5	+0: 0000
				+2: 0000	+2: 19F5
				+4: 0000	+4: 0000
				+6: 0000	+6: 0000
				(AX) = 0000	(AX) = 1A07
				(CS) = 1A0B	(CS) = 1A0B
				(DS) = 19F5	(DS) = 19F5
				(ES) = 19F5	(ES) = 19F5
				(SS) = 1A05	(SS) = 1A05
4.	0009	Mov ax,1A07(DATA)	B8071A	(SP) = 0014	(SP) = 0014
		ax,1A0/(DA1A)		Stack:	Stack:
				+0: 0000	+0: 0000
				+2: 19F5	+2: 19F5
				+4: 0000	+4: 0000
				+6: 0000	+6: 0000
				(AX) = 1A07	(AX) = 1A07
				(CS) = 1A0B	(CS) = 1A0B
				(DS) = 19F5	(DS) = 1A07
				(ES) = 19F5	(ES) = 19F5
			8ED8	(SS) = 1A05	(SS) = 1A05
5.	000C	Mov ds,ax		(SP) = 0014	(SP) = 0014
				Stack:	Stack:
				+0: 0000	+0: 0000
				+2: 19F5	+2: 19F5
				+4: 0000	+4: 0000
				+6: 0000	+6: 0000
				(AX) = 1A07	(AX) = 1A07
				(CS) = 1A0B	(CS) = 1A0B
				(DS) = 19F5	(DS) = 19F5
		Mov dx,0000		(ES) = 19F5	(ES) = 19F5
6.	000E	(OFFSET HELLO)	BA0000	(SS) = 1A05	(SS) = 1A05
		TILLO)		(SP) = 0014	(SP) = 0014
				Stack:	Stack:
				+0: 0000	+0: 0000

				+2: 19F5	+2: 19F5
				+4: 0000	+4: 0000
				+6: 0000	+6: 0000
				(AX) = 1A07	(AX) = 1A07
				(CS) = 1A0B	(CS) = 1A0B
				(DS) = 19F5	(DS) = 1A07
				(ES) = 19F5	(ES) = 19F5
				(SS) = 1A05	(SS) = 1A05
7.	0011	Call 0000	E8ECFF	(SP) = 0014	(SP) = 0012
		(WriteMsg)		Stack:	Stack:
				+0: 0000	+0: 0014
				+2: 19F5	+2: 0000
				+4: 0000	+4: 19F5
				+6: 0000	+6: 0000
				(AX) = 1A07	(AX) = 0907
				(CS) = 1A0B	(CS) = 1A0B
				(DS) = 1A07	(DS) = 1A07
				(ES) = 19F5	(ES) = 19F5
				(SS) = 1A05	(SS) = 1A05
8.	0000	Mov ah,09	B409	(SP) = 0012	(SP) = 0012
				Stack:	Stack:
				+0: 0014	+0: 0014
				+2: 0000	+2: 0000
				+4: 19F5	+4: 19F5
				+6: 0000	+6: 0000
				(AX) = 0907	(AX) = 0907
				(CS) = 1A0B	(CS) = 1A0B
				(DS) = 1A07	(DS) = 1A07
				(ES) = 19F5	(ES) = 19F5
9.	0002	Int 21	CD21	(SS) = 1A05	(SS) = 1A05
				(SP) = 0012	(SP) = 0012
				Stack:	Stack:
				+0: 0014	+0: 0014
				+2: 0000	+2: 0000

				+4: 19F5	+4: 19F5
				+6: 0000	+6: 0000
				(AX) = 0907	(AX) = 0907
				(CS) = 1A0B	(CS) = 1A0B
				(DS) = 1A07	(DS) = 1A07
				(ES) = 19F5	(ES) = 19F5
				(SS) = 1A05	(SS) = 1A05
10.	0004	ret	С3	(SP) = 0012	(SP) = 0014
				Stack:	Stack:
				+0: 0014	+0: 0000
				+2: 0000	+2: 19F5
				+4: 19F5	+4: 0000
				+6: 0000	+6: 0000
				(AX) = 0907	(AX) = 0907
				(CS) = 1A0B	(CS) = 1A0B
				(DS) = 1A07	(DS) = 1A07
				(ES) = 19F5	(ES) = 19F5
		Mov dx, 0010		(SS) = 1A05	(SS) = 1A05
11.	0014		BA1000	(SP) = 0014	(SP) = 0014
				Stack:	Stack:
				+0: 0000	+0: 0000
				+2: 19F5	+2: 19F5
				+4: 0000	+4: 0000
				+6: 0000	+6: 0000
				(AX) = 0907	(AX) = 0907
				(CS) = 1A0B	(CS) = 1A0B
				(DS) = 1A07	(DS) = 1A07
				(ES) = 19F5	(ES) = 19F5
	001=	Q 11 0000	D05.655	(SS) = 1A05	(SS) = 1A05
12.	0017	Call 0000	E8E6FF	(SP) = 0014	(SP) = 0012
				Stack:	Stack:
				+0: 0000	+0: 001A
				+2: 19F5	+2: 0000
				+4: 0000	+4: 19F5

				+6: 0000	+6: 0000
				(AX) = 0907	(AX) = 0907
				(CS) = 1A0B	(CS) = 1A0B
				(DS) = 1A07	(DS) = 1A07
				(ES) = 19F5	(ES) = 19F5
				(SS) = 1A05	(SS) = 1A05
13.	0000	Mov ah,09	B409	(SP) = 0012	(SP) = 0012
				Stack:	Stack:
				+0: 001A	+0: 001A
				+2: 0000	+2: 0000
				+4: 19F5	+4: 19F5
				+6: 0000	+6: 0000
				(AX) = 0907	(AX) = 0907
				(CS) = 1A0B	(CS) = 1A0B
				(DS) = 1A07	(DS) = 1A07
				(ES) = 19F5	(ES) = 19F5
				(SS) = 1A05	(SS) = 1A05
14.	0002	Int 21	CD21	(SP) = 0012	(SP) = 0012
				Stack:	Stack:
				+0: 001A	+0: 001A
				+2: 0000	+2: 0000
				+4: 19F5	+4: 19F5
				+6: 0000	+6: 0000
				(AX) = 0907	(AX) = 0907
				(CS) = 1A0B	(CS) = 1A0B
				(DS) = 1A07	(DS) = 1A07
				(ES) = 19F5	(ES) = 19F5
				(SS) = 1A05	(SS) = 1A05
15.	0004	ret	С3	(SP) = 0012	(SP) = 0014
				Stack:	Stack:
				+0: 001A	+0: 0000
				+2: 0000	+2: 19F5
				+4: 19F5	+4: 0000
				+6: 0000	+6: 0000

16.	001A	Ret far	СВ	(AX) = 0907 (CS) = 1A0B (DS) = 1A07 (ES) = 19F5 (SS) = 1A05 (SP) = 0014 Stack:	(AX) = 0907 (CS) = 19F5 (DS) = 1A07 (ES) = 19F5 (SS) = 1A05 (SP) = 0018 Stack:
				+0: 0000 +2: 19F5 +4: 0000 +6: 0000	+0: 0000 +2: 0000 +4: 0000 +6: 0000
17.	0000	Int 20	CD20	(AX) = 0907 (CS) = 19F5 (DS) = 1A07 (ES) = 19F5 (SS) = 1A05 (SP) = 0018 Stack: +0: 0000 +2: 0000 +4: 0000 +6: 0000	end

Результат работы программы hello2.exe

```
C:\>hello2.exe
Hello Worlds!
Student from 0382 - Danil Didenko
C:\>
```

Выводы.

Выполнены трансляция, отладка и запуск программы на языке ассемблер.

ПРИЛОЖЕНИЕ А ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: hello1.asm

```
DOSSEG
                              ; Задание сегментов под ДОС
        .MODEL SMALL
                             ; Модель памяти-SMALL(Малая)
        .STACK 100h
                            ; Отвести под Стек 256 байт
        .DATA
                            ; Начало сегмента данных
     Greeting LABEL BYTE
                                   ; Текст приветствия
       DB 'Hi Im student Danil Didenko',13,10,'$'
       . CODE
                        ; Начало сегмента кода
       mov ax, @data
                              ; Загрузка в DS адреса начала
       mov ds, ax
                             ; сегмента данных
       mov dx, OFFSET Greeting
                                     ; Загрузка в dx смещения
                                          ; адреса текста
приветствия
     DisplayGreeting:
       mov ah, 9
                            ; # функции ДОС печати строки
       int 21h
                            ; вывод на экран приветствия
       mov ah, 4ch
                             ; # функции ДОС завершения программы
       int 21h ; завершение программы и выход в ДОС
  END
Название файла: hello1.lst
Microsoft (R) Macro Assembler Version 5.10
                                                        9/15/21
08:46:47
                                               Page 1-1
                       DOSSEG
                       .MODEL SMALL
                       .STACK 100h
                       .DATA
      0000
                         Greeting LABEL BYTE
      0000 48 69 20 49 6D 20 DB 'Hi Im student Danil
Didenko',13,10,'$'
           73 74 75 64 65 6E
           74 20 44 61 6E 69
```

6C 20 44 69 64 65

6E 6B 6F 0D 0A 24

				. COL	DΕ						
	0000	В8		R	mov	ax,	@dat	a			
	0003	8E	D8		1	mov	ds, a	ЭX			
	0005	BA	0000	R	mov	dx,	OFFS	ET Gr	eeting	J	
	0008			D.	isplay		_				
	0008						ah,				
	000A						211				
	000C							, 4ch			
	000E	CD	21	END		INL	211	1			
				END							
	Micros	oft	(R) I	Macro Asse	embler	· Vers	ion	5.10			9/15/21
08:46			(, -								·, - ·,
Symbol	ls-1										
	Segmen	ts d	and G	roups:							
				$N\ a\ m\ e$		I	engt	h	Alig	gn	Combine
Class											
						• •		GROUI			
	_DAT.					• •		OOLE	WORD	PUBLI	C
	'DATA'							0100	D 71 D 71	CTI A CT	'STACK'
	TEXT									PUBLI	
-	- 'EAT 'CODE'					• •		0010	WORD	РОБЦІ	C
	CODE										
	Symbol.	s:									
	1										
				$N\ a\ m\ e$		I	'уре	Valu	ıe	Attr	
	DISPLA	YGR	EETIN	G				L NEA	AR	0008	_TEXT
	GREETI	NG						L BYT	TE.	0000	_DATA

TEXT $_TEXT$

```
TEXT 0
   TEXT 0101h
   TEXT 0
   TEXT hello1
   TEXT 510
       18 Source Lines
       18 Total Lines
       19 Symbols
     47994 + 463361 Bytes symbol space free
        0 Warning Errors
    O Severe Errors
Название файла: hello2.asm
   ; HELLO2 - Учебная программа N2 лаб.раб.#1 по дисциплине
"Архитектура компьютера"
   ;Программа использует процедуру для печати строки
   ;TEKCT ПРОГРАММЫ
   EOFLine EQU '$'; Определение символьной константы
                      ;"Конец строки"
   ; Стек программы
```

ASSUME CS:CODE, SS:AStack

AStack SEGMENT STACK

DW 12 DUP('!') ; Отводится 12 слов памяти

AStack ENDS

; Данные программы

DATA SEGMENT

; Директивы описания данных

```
GREETING DB 'Student from 0382 - Danil Didenko$'
     DATA
              ENDS
     ; Код программы
     CODE
               SEGMENT
     ; Процедура печати строки
     WriteMsg PROC NEAR
                    AH,9
               mov
               int 21h ; Вызов функции DOS по прерыванию
               ret
     WriteMsg ENDP
     ; Головная процедура
     Main
               PROC
                    FAR
               push DS
                              ;\ Сохранение адреса начала PSP в стеке
               sub
                     AX,AX
                              ; > для последующего восстановления по
               push AX
                              ;/ команде ret, завершающей процедуру.
                                         ; Загрузка сегментного
               mov
                    AX, DATA
               mov DS, AX
                                         ; регистра данных.
               mov DX, OFFSET HELLO ; Вывод на экран первой
               call WriteMsg
                                         ; строки приветствия.
                     DX, OFFSET GREETING ; Вывод на экран второй
               call WriteMsq
                                         ; строки приветствия.
               ret
                                         ; Выход в DOS по команде,
                                         ; находящейся в 1-ом слове
PSP.
     Main
              ENDP
     CODE
               ENDS
               END Main
Название файла: hello2.lst
Microsoft (R) Macro Assembler Version 5.10
                                                            9/15/21
11:28:21
                                                                  Page
1 - 1
      = 0024
                                EOFLine EQU '$'
```

HELLO DB 'Hello Worlds!', OAH, ODH, EOFLine

ASSUME CS:CODE, SS:AStack

	0000		0C[AStac	k	SEGME			ACK DUP	('!	')		
	002	_	j	ı												
								_								
	0018						AStac	K	ENDS							
	0000						DATA		SEGME	'NT'						
	0000	4.0	6.5			c-										0
	0000	48	65	6C	6C	6 <i>F</i>	20	HELL	0	DΒ	'He	110	Wo	rlds	!',	<i>OAH</i> ,
ODH,EC	rume	57	6F	72	6C	64	73									
			0A			01	, 3									
	0010					65	6E	GREE	TING	DB	'St	uder	nt	from	038.	2 -
Danil																
		74	20	66	72	6F	6D									
		20	30	33	38	32	20									
		2D	20	44	61	6E	69									
		6C	20	44	69	64	65									
		6E	6B	6F	24											
	0032						DATA		ENDS							
	0000						CODE		SEGME	'NT'						
	0000						Write	Msg	PROC	NEA	AR					
	0000	B4	09							mov		АН, 9	9			
	0002	CD	21							int		21h				
	0004	C3							ret							
	0005						Write	Msg	ENDP							

; Головная процедура

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	

Page

1-2

001B Main ENDP
001B CODE ENDS
END Main

Microsoft (R) Macro Assembler Version 5.10 9/15/21 11:28:21

Symbols-1

Segments and Groups:

	$N\ a\ m\ e$	Length	Align Combine
Class			
ASTACK .		. 0018	PARA STACK
CODE		. 001B	PARA NONE
DATA		. 0032	PARA NONE
Symbols:			
	$N\ a\ m\ e$	Type Value	e Attr
$\it EOFLINE$. NUMBEI	R 0024
GREETING		. L BYTE	E 0010 DATA
HELLO .		. L BYTE	E 0000 DATA
MAIN		F PROC	C 0005 CODE
Length =	0016		

TEXT 510

47 Source Lines

47 Total Lines

13 Symbols

47986 + 461321 Bytes symbol space free

0 Warning Errors

0 Severe Errors