

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

ОТЧЕТ
по лабораторной работе №1
по дисциплине «Организация ЭВМ и систем»
Тема: Трансляция, отладка и выполнение программ на языке
Ассемблера

Студент гр. 1383

Депрейс А.С.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

Изучить работу простейших программ на языке Ассемблера, выполнить их трансляцию, отладку и компоновку.

Задание.

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. 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
```

8. 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.

9. 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.

1. Просмотрен код программы hello1.asm и отредактирован в соответствии с личными данными.

2. Протранслирована программа hello1.asm с созданием объектного файла HELLO1.OBJ и файла листинга HELLO1.LST.

3. Скомпонован загрузочный модуль HELLO1.OBJ с созданием карты памяти HELLO1.MAP и исполняемого файла HELLO1.EXE

4. Выполнена программа HELLO1.EXE в автоматическом режиме:
“Вас приветствует ст.гр.1383 – Депрейс А.С”

5. Запущено выполнение программы HELLO1.EXE под управлением отладчика.

6. Начальное содержимое регистров:

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

Таблица 1 - результат прогона программы HELLO1.EXE в отладчике.

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0010	mov AX,1A07	B8 07 1A	(AX) = 0000 (IP) = 0010	(AX) = 1A07 (IP) = 0013
0013	mov DS,AX	8E D8	(DS) = 19F5 (IP) = 0013	(DS) = 1A07 (IP) = 0015
0015	mov DX,0000	BA 00 00	(DX) = 0000 (IP) = 0015	(DX) = 0000 (IP) = 0018
0018	mov AH,09	B4 09	(AX) = 1A07 (IP) = 0018	(AX) = 0907 (IP) = 001A
001A	int 21	CD 21	(IP) = 001A	(IP) = 001C
001C	mov AH,4C	B4 4C	(AX) = 0907 (IP) = 001C	(AX) = 4C07 (IP) = 001E
001E	int 21	CD 21	(AX) = 4C07 (CX) = 004C (DS) = 1A07 (IP) = 001E	(AX) = 0000 (CX) = 0000 (DS) = 19F5 (IP) = 0010

1. Просмотрен код программы hello2.asm и отредактирован в соответствии с личными данными.
2. Протранслирована программа hello2.asm с созданием объектного файла HELLO2.OBJ и файла листинга HELLO2.LST.
3. Скомпонован загрузочный модуль HELLO2.OBJ с создание карты памяти HELLO2.MAP и исполняемого файла HELLO2.EXE

4. Выполнена программа HELLO2.EXE в автоматическом режиме:
 “Hello Worlds!\nStudent from 1383 – Depreys A.S.”

5. Запущено выполнение программы HELLO2.EXE под управлением отладчика.

6. Начальное содержимое регистров:

CS = 1A0B; DS = 19F5; ES = 19F5; SS = 1A05;

Таблица 2 - результат прогона программы HELLO2.EXE в отладчике.

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0005	push DS	1E	(IP) = 0005 (SP) = 0018 Stack: +0 0000	(IP) = 0006 (SP) = 0016 Stack: +0 19F5
0006	sub AX,AX	2B C0	(IP) = 0006 (AX) = 0000	(IP) = 0008 (AX) = 0000
0008	push AX	50	(IP) = 0008 (SP) = 0016 Stack: +0 19F5 +2 0000	(IP) = 0009 (SP) = 0014 Stack: +0 0000 +2 19F5
0009	mov AX,1A07	B8 07 1A	(IP) = 0009 (AX) = 0000	(IP) = 000C (AX) = 1A07
000C	mov DS,AX	8E D8	(IP) = 000C (DS) = 19F5	(IP) = 000E (DS) = 1A07
000E	mov DX,0000	BA 00 00	(IP) = 000E (DX) = 0000	(IP) = 0011 (DX) = 0000

0011	call 0000	E8 EC FF	(IP) = 0011 (SP) = 0014 Stack: +0 0000 +2 19F5 +4 0000	(IP) = 0000 (SP) = 0012 Stack: +0 0014 +2 0000 +4 19F5
0000	mov AH,09	B4 09	(IP) = 0000 (AX) = 1A07	(IP) = 0002 (AX) = 0907
0002	int 21	CD 21	(IP) = 0002	(IP) = 0004
0004	ret	C3	(IP) = 0004 (SP) = 0012 Stack: +0 0014 +2 0000 +4 19F5	(IP) = 0014 (SP) = 0014 Stack: +0 0000 +2 19F5 +4 0000
0014	mov DX,0010	BA 10 00	(IP) = 0014 (DX) = 0000	(IP) = 0017 (DX) = 0010
0017	call 0000	E8 E6 FF	(IP) = 0017 (SP) = 0014 Stack: +0 0000 +2 19F5 +4 0000	(IP) = 0000 (SP) = 0012 Stack: +0 001A +2 0000 +4 19F5
0000	mov AH,09	B4 09	(IP) = 0000 (AX) = 0907	(IP) = 0002 (AX) = 0907
0002	int 21	CD 21	(IP) = 0002	(IP) = 0004

0004	ret	C3	(IP) = 0004 (SP) = 0012 Stack: +0 001A +2 0000 +4 19F5	(IP) = 001A (SP) = 0014 Stack: +0 0000 +2 19F5 +4 0000
001A	ret Far	CB	(IP) = 001A (CS) = 1A0B (SP) = 0014 Stack: +0 0000 +2 19F5	(IP) = 0000 (CS) = 19F5 (SP) = 0018 Stack: +0 0000 +2 0000
0000	int 20	CD 20	(IP) = 0000 (AX) = 0907 (CX) = 007B (DX) = 0010 (DS) = 1A07	(IP) = 0005 (AX) = 0000 (CX) = 0000 (DX) = 0000 (DS) = 19F5

Выводы.

В ходе лабораторной работы были изучены процессы трансляции, отладки и выполнения программ на языке Ассемблера.

ПРИЛОЖЕНИЕ А

ЛИСТИНГИ ПРОГРАММ

Название файла HELLO1.LST

□Microsoft (R) Macro Assembler Version 5.10

9/19/22 12:32:55

Page

1-1

```
1          DOSSEG
2          .MODEL    SMALL
3          .STACK    100h
4          .DATA
5 0000          Greeting LABEL BYTE
6 0000  82 A0 E1 20 AF E0  DB 'Вас приветствует ст.гр.1383
- Де                                     прейс А.С',13,10,'$'
7      A8 A2 A5 E2 E1 E2
8      A2 E3 A5 E2 20 E1
9      E2 2E A3 E0 2E 31
10     33 38 33 20 2D 20
11     84 A5 AF E0 A5 A9
12     E1 20 80 2E 91 0D
13     0A 24
14          .CODE
15 0000  B8 ---- R      mov  ax, @data
16 0003  8E D8          mov  ds, ax
17 0005  BA 0000 R      mov  dx, OFFSET Greeting
18 0008          DisplayGreeting:
19 0008  B4 09          mov  ah, 9
20 000A  CD 21          int  21h
21 000C  B4 4C          mov  ah, 4ch
22 000E  CD 21          int  21h
23          END
```

□Microsoft (R) Macro Assembler Version 5.10

9/19/22 12:32:55

Symbols-1

Segments and Groups:

N a m e	Length	Align	Combine	Class
DGROUP	GROUP			
_DATA	002C	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	hello1	
@VERSION	TEXT	510	

16 Source Lines

16 Total Lines

19 Symbols

47458 + 463897 Bytes symbol space free

0 Warning Errors

0 Severe Errors

Название файла HELLO2.LST

Microsoft (R) Macro Assembler Version 5.10

9/19/22 20:48:11

Page

1-1

```
1 = 0024                                EOFLine EQU '$'
2
3                                ASSUME CS:CODE, SS:AStack
4
5 0000                                AStack    SEGMENT    STACK
6 0000 000C[                            DW 12 DUP('!')
7      0021
8      ]
9
10 0018                                AStack    ENDS
11
12
13 0000                                DATA      SEGMENT
14
15
16 0000 48 65 6C 6C 6F 20HELLO          DB 'Hello Worlds!', 0AH,
0DH,
                                EOFLine
17      57 6F 72 6C 64 73
18      21 0A 0D 24
19 0010 53 74 75 64 65 6EGREETING      DB 'Student from 1383 -
Depre
                                ys A.S.$'
20      74 20 66 72 6F 6D
21      20 31 33 38 33 20
22      2D 20 44 65 70 72
23      65 79 73 20 41 2E
24      53 2E 24
25 0031                                DATA      ENDS
26
27
```

28 0000	CODE	SEGMENT	
29 0000	WriteMsg	PROC	NEAR
30 0000 B4 09		mov	AH,9
31 0002 CD 21		int	21h
32 0004 C3		ret	
33 0005	WriteMsg	ENDP	
34			
35			
36 0005	Main	PROC	FAR
37 0005 1E		push	DS
38 0006 2B C0		sub	AX,AX
39 0008 50		push	AX
40 0009 B8 ---- R		mov	AX,DATA
41 000C 8E D8		mov	DS,AX
42 000E BA 0000 R		mov	DX, OFFSET HELLO
43 0011 E8 0000 R		call	WriteMsg
44 0014 BA 0010 R		mov	DX, OFFSET GREETING
45 0017 E8 0000 R		call	WriteMsg
46 001A CB		ret	
47			
48 001B	Main	ENDP	
49 001B	CODE	ENDS	
50		END Main	

Microsoft (R) Macro Assembler Version 5.10

9/19/22 20:48:11

Symbols-1

Segments and Groups:

N a m e	Length	Align	Combine Class
ASTACK	0018	PARA	STACK
CODE	001B	PARA	NONE
DATA	0031	PARA	NONE

Symbols:

	N a m e	Type	Value	Attr
	EOFLINE	NUMBER		0024
	GREETING	L BYTE		0010 DATA
	HELLO	L BYTE		0000 DATA
0016	MAIN	F PROC		0005 CODE Length =
0005	WRITEMSG	N PROC		0000 CODE Length =
	@CPU	TEXT	0101h	
	@FILENAME	TEXT	hello2	
	@VERSION	TEXT	510	

40 Source Lines
40 Total Lines
13 Symbols

47452 + 461855 Bytes symbol space free

0 Warning Errors
0 Severe Errors

ПРИЛОЖЕНИЕ В

КОД ПРОГРАММ

Название файла HELLO1.ASM

```

; HELLO1.ASM - упрощенная версия учебной программы лаб.раб. N1
;               по дисциплине "Архитектура компьютера"
; *****
; Назначение: Программа формирует и выводит на экран приветствие
;               пользователя с помощью функции ДОС "Вывод строки"
;               (номер 09 прерывание 21h), которая:
;               - обеспечивает вывод на экран строки символов,
;               заканчивающейся знаком "$";
;               - требует задания в регистре ah номера функции=09h,
;               а в регистре dx - смещения адреса выводимой
;               строки;
;               - использует регистр ax и не сохраняет его
;               содержимое.
;
*****

                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 'Вас приветствует ст.гр.1383 - Депрейс А.С',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 HELLO2.ASM

```

; HELLO2 - Учебная программа N2 лаб.раб.#1 по дисциплине
"Архитектура компьютера"
;          Программа использует процедуру для печати строки
;
;          ТЕКСТ 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

HELLO     DB 'Hello Worlds!', 0AH, 0DH, EOFLine
GREETING  DB 'Student from 4350 - $'
DATA      ENDS

; Код программы

```

```

CODE          SEGMENT

; Процедура печати строки
WriteMsg  PROC  NEAR
            mov    AH,9
            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.
            mov    AX,DATA ; Загрузка сегментного
            mov    DS,AX ; регистра данных.
            mov    DX, OFFSET HELLO ; Вывод на экран первой
            call   WriteMsg ; строки приветствия.
            mov    DX, OFFSET GREETING ; Вывод на экран второй
            call   WriteMsg ; строки приветствия.
            ret ; Выход в DOS по команде,
                ; находящейся в 1-ом слове PSP.

Main      ENDP
CODE      ENDS
          END Main

```