

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 Ассемблере.

Студент гр. 0383

Бояркин Н.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Изучить механизм работы трансляции, отладки и выполнении программ на языке Ассемблер.

Задание.

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, по каждой из которых необходимо представить протокол с фиксацией всех выполняемых действий и полученных результатов, и подписать его у преподавателя.

Уточнение задания следует посмотреть в файле lr1_comp.txt каталога Задания.

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. 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.

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

Табл. 1

Адрес команды	Символический код команды	16-ричный 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.

Отчет по работе должен содержать:

1. текст задания;
2. тексты исходных файлов программ hello1 и hello2;
3. тексты файлов диагностических сообщений hello1.lst и hello2.lst;
4. протокол работы на компьютере, включающий основные действия по пунктам 1 - 6 и протоколы пошагового исполнения каждой из программ под управлением отладчика в виде таблицы 1 (черновики протоколов должны быть подписаны преподавателем).
5. выводы по работе.

Выполнение работы.

Часть 1. Работа с файлом hello1.asm

Выполнена протранслирование программы, компоновка загрузочного модуля и запуск программы в автоматическом режиме. Программа работает корректно.

```
C:\>hello1.exe
You are welcomed by student from the group 0383 - Boyarkin N.A.
```

Произведен запуск программы под управлением отладчика с фиксацией используемых регистров до и после выполнения каждой команды в таблицу 2.

Начальное содержимое системных регистров:

(CS) = 1A05

(DS) = 19F5

(ES) = 19F5

(SS) = 1A0C

Табл. 2

Адрес команды	Символический код команды	16-ричный 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 (DS) = 1A07	(AX) = 0907 (DS) = 1A07

			(IP) = 0018	(IP) = 001A
001A	Int 21	CD21	(AX) = 0907 (DS) = 1A07 (IP) = 001A	(AX) = 0907 (DS) = 1A07 (IP) = 14A0
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 1. Работа с файлом hello2.asm

Выполнена протранслирование программы, компоновка загрузочного модуля и запуск программы в автоматическом режиме. Программа работает корректно.

```
C:\>hello2.exe
Hello Worlds!
Student from 0383 - Boyarkin Nikita
C:\>
```

Произведен запуск программы под управлением отладчика с фиксацией используемых регистров до и после выполнения каждой команды в таблицу 3. Начальное содержимое системных регистров:

(CS) = 1A0B

(DS) = 19F5

(ES) = 19F5

(SS) = 1A05

Табл. 3

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			до выполнения	после выполнения
0005	PUSH DS	1E	(AX) = 0000	(AX) = 0000

			(DS) = 19F5 (IP) = 0005	(DS) = 19F5 (IP) = 0006
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	(AX) = 0000 (DS) = 19F5 (IP) = 0009
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	(AX) = 1A07 (DS) = 1A07 (IP) = 0000
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	(AX) = 0907 (DS) = 1A07 (IP) = 0014
0014	Mov DX, 0010	BA1000	(AX) = 0907 (DS) = 1A07 (IP) = 0014	(AX) = 0907 (DS) = 1A07 (IP) = 0017
0017	CALL 0000	E8E6FF	(AX) = 0907 (DS) = 1A07 (IP) = 0017	(AX) = 0907 (DS) = 1A07 (IP) = 0000

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	(AX) = 0907 (DS) = 1A07 (IP) = 001A
001A	RET FAR	CB	(AX) = 0907 (DS) = 1A07 (IP) = 001A	(AX) = 0907 (DS) = 1A07 (IP) = 001A
0000	Int 20	CD20	(AX) = 0907 (DS) = 1A07 (IP) = 0000	(AX) = 0000 (DS) = 19F5 (IP) = 0005

Разработанный программный код см. в приложении А.

Результаты прогона программы под управлением отладчика **hello1.asm**
(сокращенное описание сегментов):

- При таком описании требуется обязательное задание модели памяти, в условиях которой используется данная программа.

.MODEL тип_модели_памяти

Эта директива накладывает ограничения на комбинирование сегментов
(таблица)

- Greeting LABEL BYTE - определение метки типа byte
- CS: в регистр AX помещается смещение сегмента
- В регистр DX помещается значение смещение начала сообщения
- Отличие от полного описания сегментов заключается в отсутствии директивы ENDS. Таким образом, в результате создаются предопределенные переменные, которые содержат начальные адреса

сегментов: @Code, @Data, @Stack, @Const, @BBS. Следовательно можно написать:

```
mov ax, @data  
mov ds, ax
```

Результаты прогона программы под управлением отладчика **hello2.asm** (полное описание сегментов):

- Используется сегмент кода (CS) и сегмент стека (SS) для доступа к информации
- (DS): Директивы описания данных - HELLO и GREETING
- (CS): Описание процедуры печати строк
- Загрузка сегментного регистра данных, аналогичная в программе hello1.asm
- Вызов строки HELLO и GREETING
- Завершение программы

Выводы.

Был изучен механизм работы трансляции, отладки и выполнении программ на языке Ассемблер.

ПРИЛОЖЕНИЕ А

Исходный 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 'You are welcomed by a student from the group 0383 - Boyarkin
N.A.',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

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 0383 - Boyarkin Nikita $'
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
```

Название файла: hello1.lst

B2 D1 83 D0 B5 D1
 82 20 D1 81 D1 82
 2E D0 B3 D1 80 2E
 30 33 38 33 20 2D
 20 D0 91 D0 BE D1

Microsoft (R) Macro Assembler Version 5.10
 9/11/21 14:54:10
 Page
 1-2

8F D1 80 D0 BA D0
 B8 D0 BD 20 D0 9D
 2E D0 90 2E 0D 0A
 24

```

      .CODE                                ; На
      ; ало сегмента кода
0000 B8 ---- R      mov ax, @data      ;
3a
      ; грузка в DS адреса начала
0003 8E D8          mov ds, ax
; се
      ; гмента данных
0005 BA 0000 R      mov dx, OFFSET Greeting ;
3a
      ; грузка в dx смещения

0008      DisplayGreeting:
0008 B4 09          mov ah, 9
; # ф
      ; ункции ДОС печати строки
000A CD 21          int 21h
; ВЫ
      ; од на экран приветствия
000C B4 4C          mov ah, 4ch
; # ф
      ; ункции ДОС завершения про
      ; граммы
000E CD 21          int 21h
; за
      ; ершение программы и выход
      ; в ДОС
      END

```

Microsoft (R) Macro Assembler Version 5.10
 9/11/21 14:54:10
 Symbols-1

Segments and Groups:

Class	N a m e	Length	Align	Combine
DGROUP	GROUP		
_DATA	0049	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	

33 Source Lines

33 Total Lines

19 Symbols

47994 + 459266 Bytes symbol space free

0 Warning Errors

0 Severe Errors

Название файла: hello2.lst

Microsoft (R) Macro Assembler Version 5.10
9/11/21 15:05:06

Page

1-1

```
; HELLO2 - Учебная программа N2
; лаб.раб.#1 по дисциплине "
; архитектура компьютера"
; Программа использу
; ет процедуру для печати ст
; роки
;
; ТЕКСТ ПРОГРАММЫ
```

```
= 0024 EOFLine EQU '$' ; Определен
; е символьной константы
; "Конец с
; роки"
```

```

; Стек программы

ASSUME CS:CODE, SS:AStack

0000          AStack    SEGMENT    STACK
0000  000C[          DW 12 DUP('!')    ; Отводится♦
          ♦ 12 слов памяти
          0021
          ]

0018          AStack    ENDS

; Данные программы

0000          DATA      SEGMENT

; Директивы описания данн
ых

          0000  48 65 6C 6C 6F 20  HELLO          DB 'Hello Worlds!', 0AH,
0DH,EOFLine          57 6F 72 6C 64 73
          21 0A 0D 24
          0010  53 74 75 64 65 6E  GREETING      DB 'Student from 0383 -
Boyarkin Niki
          ta '$'
          74 20 66 72 6F 6D
          20 30 33 38 33 20
          2D 20 42 6F 79 61
          72 6B 69 6E 20 4E
          69 6B 69 74 61 20
          24
          0035          DATA      ENDS

; Код программы

0000          CODE      SEGMENT
; Процедура печати строки
0000          WriteMsg  PROC  NEAR
0000  B4 09          mov  AH,9
0002  CD 21          int  21h ; Вызов функции♦
          ♦ DOS по прерыванию

Microsoft      (R)      Macro      Assembler      Version      5.10
9/11/21 15:05:06

1-2

0004  C3          ret
0005          WriteMsg  ENDP

```

```

; Головная процедура
0005          Main      PROC FAR
0005 1E          push    DS          ;\ Сохранени
           е адреса начала PSP в стеке
0006 2B C0          sub    AX,AX      ; > для после♦
           ♦ующего восстановления по
0008 50          push    AX          ;/ команде ret
           , завершающей процедуру.
0009 B8 ---- R      mov     AX,DATA      ; Загр♦
           ♦зка сегментного
000C 8E D8          mov     DS,AX          ;
реги♦
           ♦тра данных.
000E BA 0000 R      mov     DX, OFFSET HELLO      ; Выво♦
           ♦ на экран первой
0011 E8 0000 R      call    WriteMsg          ; стро♦
           ♦и приветствия.
0014 BA 0010 R      mov     DX, OFFSET GREETING ; Выво♦
           ♦ на экран второй
0017 E8 0000 R      call    WriteMsg          ; стро♦
           ♦и приветствия.
001A CB          ret                      ; Выхо♦
           ♦ в DOS по команде,
           ; нахо♦
           ♦ящейся в 1-ом слове PSP.
001B          Main      ENDP
001B          CODE      ENDS
           END Main

```

Microsoft (R) Macro Assembler Version 5.10
9/11/21 15:05:06

Symbols-1

Segments and Groups:

Class	N a m e	Length	Align	Combine
ASTACK		0018	PARA	STACK
CODE		001B	PARA	NONE
DATA		0035	PARA	NONE

Symbols:

	N a m e	Type	Value	Attr
EOFLINE		NUMBER	0024	
GREETING		L BYTE	0010	DATA
HELLO		L BYTE	0000	DATA

MAIN F PROC 0005 CODE Length
= 0016

WRITEMSG N PROC 0000 CODE Length
= 0005

@CPU TEXT 0101h
@FILENAME TEXT hello2
@VERSION TEXT 510

52 Source Lines
52 Total Lines
13 Symbols

47986 + 459271 Bytes symbol space free

0 Warning Errors
0 Severe Errors