

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: Трансляция, отладка и выполнение программы
языке Ассемблера

Студент гр. 0383

Самара. Р.Д.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Изучение трансляции, отладки и выполнения программ на языке Ассемблера.

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

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.

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.

Выполнение работы.

Часть 1

1. Просмотрена программа `hello1.asm`.

2. Структура и реализация каждого сегмента программы разобраны.

Непонятные фрагменты прояснены у преподавателя. Строка-приветствие преобразована в соответствии со своими личными данными.

3. Файл `hello1.asm` из каталога Задания в каталог Masm загружен.

4. Программа протранслирована с помощью строки

```
> masm hello1.asm
```

с созданием объектного файла и файла диагностических сообщений (файла листинга). Синтаксические ошибки не обнаружены транслятором.

5. Загрузочный модуль скомпонован с помощью строки

```
> link hello1.obj
```

с созданием карты памяти и исполняемого файла `hello1.exe`.

6. Программа выполнена в автоматическом режиме путем набора строки

```
> hello1.exe
```

Программа работает корректно. Результат: «Greetings from Samara Roman, 0383»

7. Запущено выполнение программы под управлением отладчика с помощью команды

```
> afd hello1.exe (afdpro hello1.exe)
```

Начальное содержимое сегментных регистров:

(CS) = 1A05, (DS) = 19F5, (ES) = 19F5 и (SS) = 1A0A.

Выполнена программа в пошаговом режиме с фиксацией используемых регистров и ячеек памяти до и после выполнения каждой команды в таблицу 1.

Таблица 1.

Адрес Команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0010	Mov AX, 1A07	8B071A	(AX) = 0000 (IP) = 0010	(AX) = 1A07 (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 (DX) = 0000 (IP) = 0015	(AX) = 1A07 (DX) = 0000 (IP) = 0018
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 (DS) = 1A07	(AX) = 0000 (DS) = 19F5 (IP) = 0010

Часть 2. Работа с hello2.asm

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

Результат выполнения: «Hello worlds!\nStudent from 0383 – Samara R. D.»

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

Начальное содержимое сегментных регистров: (CS) = 1A0B, (DS) = 19F5, (ES) = 19F5 и (SS) = 1A05

Таблица 2.

Адрес Команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0005	PUSH DS	1E	(IP) = 0005	(IP) = 0006
0006	SUB AX, AX	2BC0	(IP) = 0006	(IP) = 0008
0008	PUSH AX	50	(SP) = 0016 (IP) = 0008	(SP) = 0014 (IP) = 0009
0009	MOV AX, 1A07	B8071A	(AX) = 0000 (IP) = 0009	(AX) = 1A07 (IP) = 000C
000C	MOV DX, AX	8ED8	(DS) = 19F5 (IP) = 000C	(DS) = 1A07 (IP) = 000E

000E	MOV DX, 0000	BA0000	(DX) = 0000 (AX) = 2D87 (IP) = 000E	(DX) = 0000 (AX) = 2D87 (IP) = 0011
0011	CALL 0000	E8ECFF	(SP) = 0014 (IP) = 0011	(SP) = 0012 (IP) = 0000
0000	MOV AH, 09	B409	(AX) = 1A07 (IP) = 0000	(AX) = 0907 (IP) = 0002
0002	INT 21	CD21	(IP) = 0002	(IP) = 0004
0004	RET	C3	(SP) = 0012 (IP) = 0004	(SP) = 0014 (IP) = 0014
0014	MOV DX, 0010	BA1000	(DX) = 0000 (IP) = 0014	(DX) = 0010 (IP) = 0017
0017	CALL 0000	E8E6FF	(SP) = 0014 (IP) = 0017	(SP) = 0012 (IP) = 0000
0000	MOV AH, 09	B409	(AX) = 0907 (IP) = 0000	(AX) = 0907 (IP) = 0002
0002	Int 21	CD21	(IP) = 0002	(IP) = 0004
0004	RET	C3	(SP) = 0012 (IP) = 0004	(SP) = 0014 (IP) = 001A
001A	RET FAR	CB	(SP) = 0014 (CS) = 1A0A (IP) = 001A	(SP) = 0018 (CS) = 19F5 (IP) = 0000
0000	INT 20	CD 20	(AX) = 0907 (DS) = 1A07 (IP) = 0000	(AX) = 0000 (DS) = 19F5 (IP) = 0005

Тексты исходных файлов программ hello1 и hello2 см. в приложении А.

Тексты файлов диагностических сообщений hello1.lst и hello2.lst см. в приложении В.

Результаты прогона программы под управлением отладчика hello1.asm

(сокращенное описание сегментов):

- ASSUME определяет какой регистр сегмента используется для доступа к информации
- 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.

- Greeting LABEL BYTE - определение метки типа byte
- CS: в регистр AX помещается смещение сегмента, в котором хранятся данные
- В регистр DX помещается значение смещение начала сообщения.

Результаты прогона программы под управлением отладчика hello2.asm

(полное описание сегментов):

- Для доступа к информации используется сегмент кода (CS) и сегмент стека (SS)
- (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.

Были изучены трансляция, отладка и выполнение программ на языке Ассемблера. Рассмотрен синтаксис программ на этом языке.

ПРИЛОЖЕНИЕ А

ТЕКСТЫ ИСХОДНЫХ ФАЙЛОВ 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 'Greetings from Samara Roman, 0383',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 – Samara Roman \$'

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

```

ПРИЛОЖЕНИЕ В

ТЕКСТЫ ФАЙЛОВ ДИАГНОСТИЧЕСКИХ СООБЩЕНИЙ

Название файла: hello1.lst

Microsoft (R) Macro Assembler Version 5.10

9/15/21 21:22:35

Page 1-1

```

1
2      ; HELLO1.ASM - гĭа®йГ-- п ŷГабЁп гзГŸ-
      ®© ĭа®Ja ¬¬л « Ÿ.a Ÿ. N1
3      ;      ĭ® ºЁбжЁĭ«Ё-Г "ЪаеЁвГЄв
      га Є®¬ĭмовГа "
4      ; *****
      *****
5      ; Ќ §- зГ-ЁГ: Ъа®Ja ¬¬ д®а¬ЁагГв Ё ŷлŷ
      ®ºЁв - нЄа - ĭаЁŷГвбвŷЁГ
6      ;      ĭ®«м§®ŷ вГ«п б ĭ®¬®ймо дГ
      -ЄжЁЁ „Һ‘ ",лŷ®º бва®ЄЁ"
7      ;      (-®¬Га 09 ĭаГалŷ -ЁГ 21h)
      , Є®в®а п:
8      ;      - ®ŸГбĭГзЁŷ Гв ŷлŷ®º -
      нЄа - бва®ЄЁ бЁ¬ŷ®«®ŷ,
9      ;      § Є -зЁŷ ойГ©бп §- Є®¬
      "$";
10     ;      - ваГŸГГв § º -Ёп ŷ аГЈЁ
      бваГ ah -®¬Га дГ-ЄжЁЁ=09h,
11     ;      ŷ аГЈЁбваГ dx - б¬Г
      йГ-Ёп ºаГб ŷлŷ®ºЁ¬®©
12     ;      бва®ЄЁ;
13     ;      - Ёбĭ®«м§ГГв аГЈЁбва ах
      Ё -Г б®еа -пГв ГЈ®
14     ;      б®ºГаĭЁ¬®Г.
15     ; *****
      *****
16
17     DOSSEG
      ; ‡ º -ЁГ бГЈ¬Г-в®ŷ ĭ®º „Һ‘
18     .MODEL SMALL
      ; Ь®ºГ«м ĭ ¬пвЁ-SMALL(Ь « п)
19     .STACK 100h
      ; ЪвŷГбвЁ ĭ®º ‘вГЄ 256 Ÿ ©в
20     .DATA

```

```

; К 3 «® ЪІЈ-І-В □ --ле
21 0000 Greeting LABEL BYTE
; 'ІЄбВ İaĖŸІвбвŸĖП
22 0000 47 72 65 65 74 69 DB 'Greetings from Samara Roman, 038
3',13,10','$'
23 6E 67 73 20 66 72
24 6F 6D 20 53 61 6D
25 61 72 61 20 52 6F
26 6D 61 6E 2C 20 30
27 33 38 33 0D 0A 24
28 .CODE

```

```

; К 3 «® ЪІЈ-І-В Є®□
29 0000 B8 ---- R mov ax, @data
; ‡ Јаг§Є Ÿ DS □аІб - 3 «
30 0003 8E D8 mov ds, ax
; ЪІЈ-І-В □ --ле
31 0005 BA 0000 R mov dx, OFFSET Greeting
; ‡ Јаг§Є Ÿ dx б-ІŸІ-ĖП
32

```

Microsoft (R) Macro Assembler Version 5.10

9/15/21 21:22:35

Page 1-2

```

; □аІб вІЄбВ İaĖŸІвбвŸĖП
33 0008 DisplayGreeting:
34 0008 B4 09 mov ah, 9
; # дг-ЄжĖĖ „Һ‘ İГз вĖ бва®ЄĖ
35 000A CD 21 int 21h
; ŸлŸ®□ - нЄа - İaĖŸІвбвŸĖП
36 000C B4 4C mov ah, 4ch
; # дг-ЄжĖĖ „Һ‘ § ŸІаиІ-ĖП İa®Ја ┐┐л
37 000E CD 21 int 21h
; § ŸІаиІ-ĖІ İa®Ја ┐┐л Ė Ÿле®□ Ÿ „Һ‘
38 END

```

Microsoft (R) Macro Assembler Version 5.10

9/15/21 21:22:35

Symbols-1

Segments and Groups:

N a m e	Length	Align	Combine	Class
DGROUP	GROUP			
_DATA	0024	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

47458 + 461849 Bytes symbol space free

0 Warning Errors

0 Severe Errors

Название файла: hello2.lst

Microsoft (R) Macro Assembler Version 5.10

9/15/21 22:21:31

Page 1-1

```
1          ; HELLO2 - “зГЎ- п İa®Ja ¬¬ N2 « Ў.a
           Ў.#1 İ® ¤ĖбжĖİ«Ė-Г "ЃaeĖБГЃBга €®¬İмoв
           Га "
2          ;      Цa®Ja ¬¬ Ėбİ®«м§ГГв İa®жГ¤Г
           аГ ¤«п İГз вĖ бBa®€Ė
3          ;
4          ;      ’...Љ’ ЦĥĥřĥЃĥĥ»
5
6 = 0024          EOFLine EQU '$'      ; ĥİaГ¤Г«Г-ĖГ
```

```

6Ë¬ÿ®«M-®© €®-бВ -ВЛ
7                               ; "Љ®-Ѓж
        бВа®€Ë"
8
9           ; 'БЃ€ Īa®Ja ¬¬Л
10
11          ASSUME CS:CODE, SS:AStack
12
13 0000          AStack  SEGMENT STACK
14 0000 000C[          DW 12 DUP(?) ; Ћвÿ®¤Ëвбп 1
        2 б«®ÿ Ī ¬пвË
15      ???
16      ]
17
18 0018          AStack  ENDS
19
20              ; ,, --ЛЃ Īa®Ja ¬¬Л
21
22 0000          DATA   SEGMENT
23
24              ; „ËaЃ€вËÿл ®ĪËб -Ëп ¤ --ле
25
26 0000 48 65 6C 6C 6F 20  HELLO  DB 'Hello Worlds!', 0AH, 0DH,
        EOFLine
27      57 6F 72 6C 64 73
28      21 0A 0D 24
29 0010 53 74 75 64 65 6E  GREETING DB 'Student from 0383 - Samar
        a R. D.$'
30      74 20 66 72 6F 6D
31      20 30 33 38 33 20
32      2D 20 53 61 6D 61
33      72 61 20 52 2E 20
34      44 2E 24
35 0031          DATA   ENDS
36
37              ; Љ®¤ Īa®Ja ¬¬Л
38
39 0000          CODE    SEGMENT
40              ; Тa®жЃЃга ĪЃ вË бВа®€Ë
41 0000          WriteMsg PROC NEAR
42 0000 B4 09          mov AH,9
43 0002 CD 21          int 21h ; ,л§®ÿ дг-€жËË DO

```


S Ĩ® ĨaГaлŷ -Ėo

44 0004 C3

ret

45 0005

WriteMsg ENDP

Microsoft (R) Macro Assembler Version 5.10

9/15/21 22:21:31

Page 1-2

46

47

; ĩ®«®ŷ- п Ĩa®жГŕga

48 0005

Main PROC FAR

49 0005 1E

push DS ;\ '®ea -Г-ĖГ

ŕaГб - з « PSP ŷ бвГĖГ

50 0006 2B C0

sub AX,AX ; > ŕ«п Ĩ®б«Гŕ

гойГJ® ŷ®ббв -®ŷ«Г-Ėп Ĩ®

51 0008 50

push AX ;/ Ė®ŕ -ŕГ re

t, § ŷГaи ойГ© Ĩa®жГŕgaг.

52 0009 B8 ---- R

mov AX,DATA ; ‡

Jaг§Ė бГJŕГ-в-®J®

53 000C 8E D8

mov DS,AX ; a

ГJĖбвa ŕ --ле.

54 000E BA 0000 R

mov DX, OFFSET HELLO ; ,

лŷ®ŕ - нCa - ĨГaŷ®©

55 0011 E8 0000 R

call WriteMsg ; б

вa®ĖĖ ĨaĖŷГвбвŷĖп.

56 0014 BA 0010 R

mov DX, OFFSET GREETING ; ,

лŷ®ŕ - нCa - ŷв®a®©

57 0017 E8 0000 R

call WriteMsg ; б

вa®ĖĖ ĨaĖŷГвбвŷĖп.

58 001A CB

ret ; ,

ле®ŕ ŷ DOS Ĩ® Ė®ŕ -ŕГ,

59

; -

е®ŕпйГ©бп ŷ 1-®ŕ б«®ŷГ PSP.

60 001B

Main ENDP

61 001B

CODE ENDS

62

END Main

Microsoft (R) Macro Assembler Version 5.10

9/15/21 22:21:31

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	
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

47466 + 461841 Bytes symbol space free

0 Warning Errors
0 Severe Errors