

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 программ на языке

Ассемблера»

Студент гр. 1383

Петров А.С.

Преподаватель

Ефремов М. А.

Санкт-Петербург

2022

Цель работы.

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

Задание на лабораторную работу.

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

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. 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.

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 и отредактирован в соответствии с личными данными код программы hello1.asm.
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 созданием исполняемого файла HELLO1.EXE
4. Выполнена программа HELLO1.EXE в автоматическом режиме:
«Вас приветствует студент группы 1383 - Петров А.С.»
5. Запущено выполнение программы HELLO1.EXE под управлением отладчика.
6. Начальное содержимое сегментных регистров:
CS = 1A05; DS = 19F5; ES = 19F5; SS = 1A0B

Таблица 1 – Результат запуска 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 8D	(DS) = 19F5 (IP) = 0013	(DS) = 1A07 (IP) = 0015
0015	Mov DX,0000	BA 00 00	(IP) = 0015	(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) = 0055 (DS) = 1A07 (IP) = 001E	(AX) = 0000 (CX) = 0000 (DS) = 19F5 (IP) = 0010

1. Просмотрен и отредактирован в соответствии с личными данными код программы hello2.asm.
2. Протранслирована программа hello2.asm с созданием объектного файла HELLO2.OBJ и файла листинга HELLO2.LST.
3. Скомпонован загрузочный модуль HELLO2.OBJ с созданием карты памяти HELLO2.MAP и созданием исполняемого файла HELLO2.EXE
4. Выполнена программа HELLO2.EXE в автоматическом режиме:
«Hello Worlds!
Student from 1383 - Petrov A. S.»
5. Запущено выполнение программы HELLO2.EXE под управлением отладчика.
6. Начальное содержимое сегментных регистров:
CS = 1A0B; DS = 19F5; ES = 19F5; SS = 1A05

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0005	PUSH DS	1E	(SP) = 0018 (IP) = 0005 Stack: +0 0000	(SP) = 0016 (IP) = 0006 Stack: +0 19F5
0006	SUB AX,AX	2B C0	(IP) = 0006	(IP) = 0008
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	(IP) = 0011
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) = 0002 (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	(SP) = 0014 (IP) = 0017 Stack: +0 0000 +2 19F5 +4 0000	(SP) = 0012 (IP) = 0000 Stack: +0 001A +2 0000 +4 19F5
0000	MOV AH,09	B4 09	(IP) = 0000	(IP) = 0002
0002	INT 21	CD 21	(IP) = 0002	(IP) = 0004
0004	RET	C3	(SP) = 0012 (IP) = 0004 Stack: +0 001A +2 0000 +4 19F5	(SP) = 0014 (IP) = 001A Stack: +0 0000 +2 19F5 +4 0000

001A	RET Far	CB	(IP) = 001A (CS) = 1A0B (SP) = 0014 Stack: +2 19F5	(IP) = 0000 (CS) = 19F5 (SP) = 0018 Stack: +2 0000
0000	INT 20	CD 20	(AX) = 0907 (CX) = 007B (DX) = 0010 (CS) = 19F5 (DS) = 1A07 (IP) = 0000	(AX) = 0000 (CX) = 0000 (DX) = 0000 (CS) = 1A0B (DS) = 19F5 (IP) = 0005

Выводы.

В ходе выполнения работы был получен навыки трансляции отладки и запуска программ на языке Ассемблера.

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
DOS
.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 студент группы 1383 - Петров 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 1383 - Petrov A. S.$'
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 Б

ЛИСТИНГ ПРОГРАММ

Название файла hello1.lst

Microsoft (R) Macro Assembler Version 5.10
17:08:15

9/27/22

Page

1-1

```
1
2           ; HELLO1.ASM - упрощенная версия учебн
ой программы лаб.раб. N1
3           ;           по дисциплине "Архитект
ура компьютера"
4           ; *****
*****
5           ; Назначение: Программа формирует и выв
одит на экран приветствие
6           ;           пользователя с помощью фу
нкции ДОС "Вывод строки"
7           ;           (номер 09 прерывание 21h)
, которая:
8           ;           - обеспечивает вывод на
экран строки символов,
9           ;           заканчивающейся знаком
"$";
10          ;           - требует задания в реги
стре ah номера функции=09h,
11          ;           а в регистре dx - сме
щения адреса выводимой
12          ;           строки;
13          ;           - использует регистр ax
и не сохраняет его
14          ;           содержимое.
15          ; *****
*****
16
17          DOSSEG
           ; Задание сегментов под ДОС
18          .MODEL  SMALL
           ; Модель памяти-SMALL (Малая)
19          .STACK  100h
           ; Отвести под Стек 256 байт
20          .DATA
           ; Начало сегмента данных
21 0000      Greeting LABEL BYTE
           ; Текст приветствия
22 0000 82 A0 E1 20 AF E0 DB 'Вас приветствует студент
группы
1383 - Петров А.С.',13,10,'$'
```

```

23      A8 A2 A5 E2 E1 E2
24      A2 E3 A5 E2 20 E1
25      E2 E3 A4 A5 AD E2
26      20 A3 E0 E3 AF AF
27      EB 20 31 33 38 33
28      20 2D 20 8F A5 E2
29      E0 AE A2 20 80 2E
30      91 2E 0D 0A 24
31      .CODE
          ; Начало сегмента кода
32 0000  B8 ---- R          mov  ax, @data
          ; Загрузка в DS адреса начала
33 0003  8E D8              mov  ds, ax
          ; сегмента данных

```

Microsoft (R) Macro Assembler Version 5.10
17:08:15

9/27/22

Page

1-2

```

34 0005  BA 0000 R          mov  dx, OFFSET Greeting
          ; Загрузка в dx смещения
35
          ; адреса текста приветствия
36 0008                      DisplayGreeting:
37 0008  B4 09              mov  ah, 9
          ; # функции ДОС печати строки
38 000A  CD 21              int  21h
          ; вывод на экран приветствия
39 000C  B4 4C              mov  ah, 4ch
          ; # функции ДОС завершения программы
40 000E  CD 21              int  21h
          ; завершение программы и выход в ДОС
41                      END

```

Microsoft (R) Macro Assembler Version 5.10
17:08:15

9/27/22

Symbols-1

Segments and Groups:

	N a m e	Length	Align	Combine	Class
DGROUP	GROUP			
_DATA	0035	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
18:51:07

9/27/22

Page

1-1

```

1          ; HELLO2 - Учебная програ
;ма N2  лаб.раб.#1 по дис
;иплине "Архитектура
;омпьютера"
2          ;          Программа испо
;ьзует процедуру для
;ечати строки
3          ;
4          ;          ТЕКСТ  ПРОГРАММЫ
5
6 = 0024          EOFLine  EQU  '$'          ; Опре
;ление символьной кон
станты
7          ;          "Кон
ещ строки"
8
9          ; Стек  программы
10
11         ASSUME CS:CODE, SS:Astack
12
13 0000          Astack  SEGMENT  STACK
14 0000  000C[          DW 12 DUP('!!')          ; Отво

```

```

        ?ится 12 слов памяти
15      0021
16      ]
17
18 0018      AStack      ENDS
19
20      ; Данные программы
21
22 0000      DATA      SEGMENT
23
24      ; Директивы описания
        данных
25
26 0000  48 65 6C 6C 6F 20HELLO      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 6EGREETING  DB 'Student from 1383 -
Petro
        v A. S.$'
30      74 20 66 72 6F 6D
31      20 31 33 38 33 20
32      2D 20 50 65 74 72
33      6F 76 20 41 2E 20
34      53 2E 24
35 0031      DATA      ENDS
36
37      ; Код программы
38
39 0000      CODE      SEGMENT
40      ; Процедура печати ст?
        ?оки
41 0000      WriteMsg  PROC  NEAR
Microsoft (R) Macro Assembler Version 5.10      9/27/22
18:51:07

```

Page

1-2

```

42 0000  B4 09      mov  AH,9
43 0002  CD 21      int  21h ; Вызов фу?
        ?кции DOS по прерыванию
44 0004  C3      ret
45 0005      WriteMsg  ENDP
46
47      ; Головная процедура
48 0005      Main      PROC  FAR
49 0005  1E      push  DS      ; \ Сохра
        нение адреса начала PS
        Р в стеке
50 0006  2B C0      sub   AX,AX      ; > для п?

```

```

        ?следующего восстано?
        ?ления по
51 0008 50                push  AX          ;/  коман
        де ret, завершающей про
        цедуру.
52 0009 B8 ---- R        mov  AX,DATA      ;
?
        ?агрузка сегментного
53 000C 8E D8            mov  DS,AX        ;
?
        ?егистра данных.
54 000E BA 0000 R        mov  DX, OFFSET HELLO ;
?
        ?ывод на экран первой
55 0011 E8 0000 R        call WriteMsg     ;
?
        ?троки приветствия.
56 0014 BA 0010 R        mov  DX, OFFSET GREETING ;
?
        ?ывод на экран второй
57 0017 E8 0000 R        call WriteMsg     ;
?
        ?троки приветствия.
58 001A CB              ret                ;
?
        ?ыход в DOS по команде,
59                                ; ?
        ?аходящейся в 1-ом сло?
        ?е PSP.
60 001B                Main      ENDP
61 001B                CODE      ENDS
62                                END Main

```

Microsoft (R) Macro Assembler Version 5.10 9/27/22
 18:51:07

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 0016	F PROC	0005	CODE	Length =
WRITEMSG 0005	N PROC	0000	CODE	Length =
@CPU	TEXT	0101h		
@FILENAME	TEXT	hello2		
@VERSION	TEXT	510		

52 Source Lines
 52 Total Lines
 13 Symbols

47450 + 459807 Bytes symbol space free

0 Warning Errors
 0 Severe Errors