

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

ОТЧЁТ

по лабораторной работе №1

по дисциплине «Организация ЭВМ и систем»

**Тема: Освоение трансляции, выполнения и отладки программ на
языке Ассемблера процессора Intel x86.**

Студент гр. 0381

Ковалев П. А.

Преподаватель

Ефремов М. А.

Санкт-Петербург

2022

Цель работы

Освоение трансляции, выполнения и отладки программ на языке Ассемблера процессора Intel x86.

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. 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.obj
```

Выполнение работы

1. Просмотрен текст программы `hello1.asm`, строка-приветствие преобразована в соответствии с личными данными

```
...  
DB 'Вас приветствует ст.гр.1383 – Ковалев П. А.',13,10,'$'  
...
```

Файл сохранен в кодировке `cp866`.

2. Запущен эмулятор `DosBox` командой

```
$ dosbox .
```

Для вывода русских символов в эмуляторе запущена команда

```
> keyb ru 866
```

3. В окружении `DosBox` выполнена трансляция программы `hello1.asm` из ассемблера в объектный код.

```
> nasm hello1.asm , , , ,
```

Ошибки трансляции не были обнаружены.

4. Далее модуль был скомпонован в исполняемый файл

```
> link hello1.obj , , , , ,
```

5. Запущено выполнение программы HELLO1.EXE

```
> HELLO1.EXE
```

Вас приветствует ст.гр.1383 – Ковалев П. А.

Результат выполнения программы HELLO1.EXE корректен.

6. Запущено выполнение программы под управлением отладчика

```
> afdpro HELLO1.EXE
```

Результат выполнения программы HELLO1.EXE представлен в таблице (1).

7. Просмотрен текст программы hello2.asm, строка-приветствие преобразована в соответствии с личными данными

```
...
```

```
GREETING DB 'Student from 1383 – Kovalev P. A.$'
```

```
...
```

8. Выполнена трансляция текста программы в объектный код, затем объектный код был скомпилирован в исполняемый файл

```
> nasm hello2.asm , , , ,
```

```
> link hello2.obj , , , , ,
```

9. Запущено выполнение программы HELLO2.EXE

```
> HELLO2.EXE
```

Hello Worlds!

Student from 1383 – Kovalev P. A.

Результат выполнения программы HELLO2.EXE корректен.

10. Запущено выполнение программы под управлением отладчика

```
> afdpro HELLO2.EXE
```

Результат выполнения программы HELLO2 .EXE представлен в таблице (2).

Вывод

Была освоена трансляция, выполнение и отладка программ на языке Ассемблера процессора Intel x86.

Приложение А

Исходный код программ

Имя файла: hello1.asm

```
; HELLO1.ASM - упрощенная версия учебной программы лабраб.. 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-SMALLМалая()
.STACK 100h                 ; Отвести под Стек 256 байт
.DATA                       ; Начало сегмента данных
Greeting LABEL BYTE         ; Текст приветствия
DB 'Вас' приветствует стгр..1383 - Ковалев П. А.',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
```

Имя файла: hello1.lst

Microsoft (R) Macro Assembler Version 5.10

9/27/22 22:55:47

Page 1-1

```
17          DOSSEG
18          .MODEL SMALL
19          .STACK 100h
20          .DATA
```



```

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

```

47464 + 459796 Bytes symbol space free

```

0 Warning Errors
0 Severe Errors

```

Имя файла: hello1.map

Start	Stop	Length	Name	Class
00000H	0001FH	00020H	_TEXT	CODE
00020H	00069H	0004AH	_DATA	DATA
00070H	0016FH	00100H	STACK	STACK

```

Origin Group
0002:0 DGROUP

```

Имя файла: hello2.asm

```

; HELLO2 - Учебная программа N2 лабраб..#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 - Kovalev P. A.$'
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 01- слове PSP.

Main        ENDP
CODE        ENDS
            END Main
```

Имя файла: hello2.lst

Microsoft (R) Macro Assembler Version 5.10

9/28/22 00:00:59

Page 1-1

= 0024 EOFLine EQU '\$'

ASSUME CS:CODE, SS:AStack

```
0000                    AStack    SEGMENT   STACK
0000   000C[                                    DW 12 DUP('!')
0021                    ]
```

```

0018          AStack      ENDS

0000          DATA      SEGMENT

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 1383 - Kovalev P. A.
      $'
      74 20 66 72 6F 6D
      20 31 33 38 33 20
      2D 20 4B 6F 76 61
      6C 65 76 20 50 2E
      20 41 2E 24
0032          DATA      ENDS

0000          CODE      SEGMENT
0000          WriteMsg    PROC      NEAR
0000  B4 09                      mov     AH,9
0002  CD 21                      int     21h
0004  C3                          ret
Microsoft (R) Macro Assembler Version 5.10
9/28/22 00:00:59
Page      1-2

```

```

0005          WriteMsg    ENDP

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
001B          Main        ENDP
001B          CODE      ENDS
                        END Main
Microsoft (R) Macro Assembler Version 5.10
9/28/22 00:00:59
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	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

47988 + 459269 Bytes symbol space free

0 Warning Errors

0 Severe Errors

Имя файла: hello2.map

Start	Stop	Length	Name	Class
00000H	00017H	00018H	ASTACK	
00020H	00051H	00032H	DATA	
00060H	0007AH	0001BH	CODE	

Program entry point at 0006:0005

Таблица 1: Результаты выполнения программы HELLO1 .EXE в пошаговом режиме

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров до выполнения команды	Содержимое регистров после выполнения команды
0010	MOV AX,1A07	B8071A	AX=0000 IP=0010	AX=1A07 IP=0013
0013	MOV DS,AX	8ED8	DS=19F5 IP=0013	DS=1A07 IP=0015
0015	MOV DX,0000	BA0000	DX=0000 IP=0015	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	IP=001E	IP=0020

Таблица 2: Результаты выполнения программы HELLO2 .EXE в пошаговом режиме

Адрес коман-ды	Символический код команды	16-ричный код ко-манды	Содержимое регистров до выполнения команды	Содержимое регистров после выполнения команды	Содержимое стека до выполнения команды	Содержимое стека после выполнения команды
0005	PUSH DS	1E	SP=0018 IP=0005	SP=0016 IP=0006	+0 0000 +2 0000 +4 0000 +6 0000	+0 19F5 +2 0000 +4 0000 +6 0000
0006	SUB AX,AX	2BC0	AX=0000 IP=0006	AX=0000 IP=0008		
0008	PUSH AX	50	SP=0016 IP=0008	SP=0014 IP=0009	+0 19F5 +2 0000 +4 0000 +6 0000	+0 0000 +2 19F5 +4 0000 +6 0000
0009	MOV AX,1A07	B8071A	AX=0000 IP=0009	AX=1A07 IP=000C		
000C	MOV DS,AX	8ED8	DS=19F5 IP=000C	DS=1A07 IP=000E		
000E	MOV DX,0000	B80000	DX=0000 IP=000E	DX=0000 IP=0011		
0011	CALL 0000	E8ECFF	IP=0011 SP=0014	IP=0000 SP=0012	+0 0000 +2 19F5 +4 0000 +6 0000	+0 0014 +2 0000 +4 19F5 +6 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	IP=0004 SP=0012	IP=0014 SP=0014	+0 0014 +2 0000 +4 19F5 +6 0000	+0 0000 +2 19F5 +4 0000 +6 0000
0014	MOV DX,0010	BA1000	DX=0000 IP=0014	DX=0010 IP=0017		
0017	CALL 0000	E8ECFF	IP=0017 SP=0014	IP=0000 SP=0012	+0 0000 +2 19F5 +4 0000 +6 0000	+0 0017 +2 0000 +4 19F5 +6 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	IP=0004 SP=0012	IP=0017 SP=0014	+0 0017 +2 0000 +4 19F5 +6 0000	+0 0000 +2 19F5 +4 0000 +6 0000
0017	RET FAR	CB	IP=0017 SP=0014	IP=0000 SP=0018	+0 0000 +2 19F5 +4 0000 +6 0000	+0 0000 +2 0000 +4 0000 +6 0000