

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

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

Студент гр. 0383

Коротков А.В.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Ознакомиться с процессом трансляции, отладкой и запуском программ на языке Ассемблера

Основные теоретические положения.

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 4
```

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.

Выполнение работы:

Часть 1:

- 1) Был просмотрен и проанализирован код программы hello1.asm
- 2) Программа была протранслирована с помощью директивы `masm` с созданием файла листинга. Ошибок обнаружено не было.
- 3) Загрузочный модуль была скомпилирована с помощью директивы `link`
- 4) Программа корректно выполнена в автоматическом режиме
- 5) Программа была запущена под управлением отладчика `afddpro`.

Результаты прогона программы представлены в таблице 1.

Начальные значения: (CS)=1A05, (DS)=19F5, (ES)=19FS,
(SS)=1A0C Таблица 1.

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти до выполнения	Содержимое регистров и ячеек памяти после выполнения
0010	MOV AX, 1A07	B8071A	(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	(DX)=0000 (IP)=0015	(DX)=0000 (IP)=0018
0018	MOV AH, 09	B409	(AX)=1A07	(AX)=0907

			(IP)=0018	(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)=0010

Часть 2:

По аналогии с частью 1 настоящего задания были выполнены компиляция, линковка и запуск программы hello2.asm под управлением отладчика afdpro. Результаты прогона программы hello2.asm представлены в таблице 2.

Результат работы программы:

“Hello Worlds! \n Student from 0383 - Korotkov A.V.”

Начальные значения: (CS)=1A0B, (DS)=19F5, (ES)=19F5, (SS)=1A05 Таблица 2.

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти до выполнения	Содержимое регистров и ячеек памяти после выполнения
0005	PUSH DS	1E	(SP)=0018 (IP)=0005 Stack: +0 0000	(SP)=0016 (IP)=0006 Stack: +0 19F5
0006	SUB AX, AX	2BC0	(AX)=0000 (IP)=0006 Stack: +0 19F5	(AX)=0000 (IP)=0015 Stack: +0 19F5
0008	PUSH AX	50	(SP)=0016 (IP)=0008 Stack: +0 19F5 Stack: +2 0000	(SP)=0014 (IP)=0009 Stack: +0 0000 Stack: +2 19F5

0009	MOV AX,1A07	B8071A	AX)=0000 (IP)=0009 Stack: +0 0000 Stack: +2 19F5	(AX)=1A07 (IP)=000C Stack: +0 0000 Stack: +2 19F5
000C	MOV DS, AX	BED8	(DS)=19F5 (AX)=1A07 (IP)=000C Stack: +0 0000 Stack: +2 19F5	(DS)=1A07 (AX)=1A07 (IP)=000E Stack: +0 0000 Stack: +2 19F5
000E	MOV DX, 0000	BA0000	(DX)=0000 (IP)=000E Stack: +0 0000 Stack: +2 19F5	(DX)=0000 (IP)=0011 Stack: +0 0000 Stack: +2 19F5
0011	CALL 0000	E8ECFF	(SP)=0014 (IP) = 0011 Stack: +0 0000 Stack: +2 19F5 Stack: +4 0000	(SP)=0012 (IP) = 0000 Stack: +0 0014 Stack: +2 0000 Stack: +4 19F5

0000	MOV AH,9	B409	(AX)=1A07 (IP)=0000 Stack: +0 0014 Stack: +2 0000 Stack: +4 19F5	(AX)=0907 (IP)=0002 Stack: +0 0014 Stack: +2 0000 Stack: +4 19F5
0002	INT 21	CD21	(IP)=0002 Stack: +0 0014 Stack: +2 0000 Stack: +4 19F5	(IP)=0004 Stack: +0 0014 Stack: +2 0000 Stack: +4 19F5
0004	RET	C3	(IP)=0004 (SP)=0012 Stack: +0 0014 Stack: +2 0000 Stack: +4 19F5	(IP)=0014 (SP)=0014 Stack: +0 0000 Stack: +2 19F5 Stack: +4 0000
0014	MOV DX, 0010	BA1000	(DX)=0000 (IP)=0014 Stack: +0 0000 Stack: +2 19F5	(DX)=0010 (IP)=0017 Stack: +0 0000 Stack: +2 19F5
0017	CALL 0000	E6FF	(SP)=0014	(SP)=0012

			(IP)=0017 Stack: +0 0000 Stack: +2 19F5 Stack: +4 0000	(IP)=0000 Stack: +0 001A Stack: +2 0000 Stack: +4 19F5
0000	MOV AH,9	B409	(AX)=1A07 (IP)=0000 Stack: +0 001A Stack: +2 0000 Stack: +4 19F5	(AX)=0907 (IP)=0002 Stack: +0 001A Stack: +2 0000 Stack: +4 19F5
0002	INT 21	CD21	(IP)=0002 Stack: +0 001A Stack: +2 0000 Stack: +4 19F5	(IP)=0004 Stack: +0 001A Stack: +2 0000 Stack: +4 19F5
0004	RET	C3	(IP)=0004 (SP)=0012 Stack: +0 001A Stack: +2 0000 Stack: +4 19F5	(IP)=001A (SP)=0014 Stack: +0 0000 Stack: +2 19F5 Stack: +4 0000
001A	RET Far	CB	(IP) = 001A (SP) = 0014 (CS) = 1A0B Stack: +0 0000 Stack: +2 19F5 Stack: +4 0000	(IP) = 0000 (SP) = 0018 (CS) = 19F5 Stack: +0 0000 Stack: +2 0000 Stack: +4 0000
0000	INT 20	CD 20	(AX)=0907 (IP)=0000 (SP)=0018 (CX)=006B (CS)=19F5 (DS)=1A07 Stack: +0 0000 Stack: +2 0000	Завершение программы

Выводы.

В ходе выполнения лабораторной работы были изучены процессы компиляции файла на языке Ассемблера компилятором MASM и работы с отладчиком AFDPRO, а также изучен синтаксис языка.

ПРИЛОЖЕНИЕ А

Исходный код программы

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.
; *****
;
; mov ah, 9
DOSSEG int 21h
; mov ah, 4ch
ДОС программы
; int 21h
.MODEL SMALL в ДОС
; END
памяти-SMALL (Малая)
;
; .STACK 100h
;
; байт
;
; .DATA
;
; Greeting LABEL BYTE
;
; DB 'Вас приветствует
; ст.гр.0383 -
;
; .CODE
;
; mov ax, @data
; mov ds, ax
; mov dx, OFFSET
; Greeting
DisplayGreeting:
```

```

; Задание сегментов под
; сегмента данных
; Модель
; Загрузка в dx смещения
; Отвести под стек 256
; адреса текста приветствия
; Начало сегмента данных
; # функции ДОС печати строки
; Текст приветствия
; вывод на экран приветствия
Коротков А. В. ',13,10','$'
; # функции ДОС завершения
; Начало сегмента кода
; завершение программы и выход
; Загрузка в DS адреса начала

```

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
; Отводится 12 слов памяти
DW 12 DUP('!')

AStack ENDS

; Данные программы
DATA SEGMENT

; Директивы описания данных
HELLO DB 'Hello Worlds!', 0AH, 0DH, EOFLine
GREETING DB 'Student from 4350 - Korotkov A.V.
$'
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,
Main      ENDP          ; находящейся в 1-ом слове
                        PSP.
```

CODE ENDS

END Main

Название файла: hello1.lst

Microsoft (R) Macro Assembler Version 5.10
20:48:39

9/12/21

Page

1-1

```
;      HELLO1.ASM - упрощенная верси
я учебной программы лаб.р
б. N1
;                  по дисциплине "Ар
хитектура компьютера"

;  ****

*****

; Назначение: Программа фо
рмирует и выводит на экра
приветствие
;                  пользователя с по
ощью функции ДОС "Вывод с
роки"
;                  (номер 09 прерывани
е 21h), которая:
;                  - обеспечивает вы
од на экран строки символ
ОВ,
;                  заканчивающейся
знаком "$";
;                  - требует задания
в регистре ah номера функц
и=09h,
;                  а в регистре dx -
мещения адреса выводимо
й
;                  строки;
;                  - использует реги
трах и не сохраняет его
;                  содержимое.
;  ****

*****

DOSSEG

; Задание сегментов под ДО

С

.MODEL  SMALL
```

```

; Модель памяти-SMALL (Малая)

.STACK 100h

; Отвести под Стек 256 байт

.DATA
; Начало сегмента данных

0000 Greeting LABEL BYTE

; Текст приветствия
0000 DB 'Вас приветствует ст.г'
      .0383 - Коротков А. В.',13,10,'$'
      20 D0 BF D1 80 D0
      B8 D0 B2 D0 B5 D1
      82 D1 81 D1 82 D0
      B2 D1 83 D0 B5 D1
      82 20 D1 81 D1 82
      2E D0 B3 D1 80 2E
      30 33 38 33 20 2D
Microsoft (R) Macro Assembler Version 5.10
20:48:39
1-2
9/12/21
Page

      20 D0 9A D0 BE D1

      80 D0 BE D1 82 D0
      BA D0 BE D0 B2 20
      D0 90 2E 20 D0 92
      2E 0D 0A 24 .CODE ; На

; ало сегмента кода
0000 B8 ---- mov ax, @data ;
      R

Загрузка в DS адреса начала

0003 8E D8 mov ds, ax ;
се гмента данных

0005 BA 0000 mov dx, OFFSET Greeting ;
      R

Загрузка в dx смещения ; ад

; еса текста приветствия

0008 DisplayGreeting:
0008 B4 09 mov ah, 9 ;
#

Функции DOS печати строки

000A CD 21 int 21h ;
вы ; од на экран приветствия

```

```

000C   B4 4C                               mov  ah, 4ch                ;
                                           #
Ф      функции ДОС завершения про
000E   CD 21                               граммы  int  21h                ;
за❖      ❖ершение программы и выход❖
                                           ❖ВДОС
                                           END

```

Microsoft (R) Macro Assembler Version 5.10
20:48:39

9/12/21
Symbols-1

Segments and Groups:

N a m e	Length	Align	Combine
Class			
DGROUP.....	GROUP		
_DATA	004C WORD PUBLIC		'DATA'
STACK	0100 PARA STACK		'STACK'
_TEXT	0010 WORD PUBLIC		'CODE'

Symbols:

N a m e	Type	Value	Attr
DISPLAYGREETING	L	0008	_TEXT
	NEAR		
GREETING.....	L	0000	_DATA
	BYTE		
@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

9/12/21

1-1

```

; HELLO2 - Учебная программа N2
; лаб.раб.#1 по дисциплине "Архитектура компьютера"
; Программа использует процедуру для печати строки
;
; ТЕКСТ ПРОГРАММЫ
=      EOFLine EQU '$' ; Определен
0024

; Символьная константа
; строки" ; "Конец строки"

; Стек программы

ASSUME CS:CODE, SS:ASStack

0000 ASStack SEGMENT STACK
0000 000C[ DW 12 DUP('!') ;
; Отводится
; 12 слов памяти
0021
]
0018 ASStack END
S

; Данные программы

0000 DATA SEGMENT

; Директивы описания данных

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 4350 -
A.V. $'
74 20 66 72 6F 6D
20 34 33 35 30 20
2D 20 4B 6F 72 6F
74 6B 6F 76 20 41
0033 2E 56 2E 20 24 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
0004 C3 ret
Microsoft (R) Macro Assembler Version 5.10 9/12/21
21:18:18
1-2

0005 WriteMsg ENDP

; Головная процедура

0005 Main PROC FAR
0005 1E push DS ; \ Сохранени
е адреса PSP в стеке
начала

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 9/12/21

```


21:18:18

Symbols-
1

Segments and Groups:

Class	N a m e	Length	Align	Combine
ASTACK.....		0018	PARA	STACK
CODE.....		001B	PARA	NONE
DATA.....		0033	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