

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

ОТЧЕТ
по лабораторной работе №1
по дисциплине «Организация ЭВМ и систем»
Тема: Трансляции, отладка и выполнение программ на
языке Ассемблера.

Студент гр. 9383

Орлов Д.С.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

Цель работы.

Научиться работать с эмулятором dosbox. Освоить навыки трансляции отладки программы на Ассемблере.

Задание.

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.

Разобраться в структуре и реализации каждого сегмента программы. Непонятные фрагменты прояснить у преподавателя. Строку-приветствие преобразовать в соответствии со своими личными данными.

1. Загрузить файл hello1.asm из каталога Задания в каталог Masm.

2. 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.

3. Скомпоновать загрузочный модуль с помощью строки

```
> link hello1.obj
```

с созданием карты памяти и исполняемого файла hello1.exe.

4. Выполнить программу в автоматическом режиме путем набора строки

```
> hello1.exe
```

убедиться в корректности ее работы и зафиксировать результат выполнения в протоколе.

5. Запустить выполнение программы под управлением отладчика с помощью команды

> afd hello1.exe

Ход работы.

Часть 1.

1. Загружены файлы из каталога /tools в каталог F:\MASM_EXE
2. Запущена программа DOSBox, смонтирован виртуальный диск I: в каталоге F:\MASM_EXE при помощи mount i f:\masm_exe
3. Просмотрен код программ hello1.asm и hello2.asm, изучены структура и реализация сегментов программы. Строка-приветствие преобразована в соответствии с личными данными.
4. В DOS осуществлен переход на виртуальный диск при помощи команды i:
5. Протранслирована программа с помощью строки:

> masm hello1.asm

Создается объектный файл HELLO1.obj с сообщением об отсутствии ошибок и предупреждений.

6. Скомпонован загрузочный модуль HELLO1.exe с помощью строки:

> link hello1.obj

Линковщик создает загрузочный модуль HELLO1.exe.

7. Загружена русская кодовая таблица символов путём набора строки:

> keyb ru 866

8. Запущена программа в автоматическом режиме путем набора строки:

> hello1.exe

9. Вывод программы:

> Вас приветствует ст.гр.9383 — Орлов Д. С.

10. Выполнен запуск программы HELLO1.exe в пошаговом режиме с фиксацией используемых регистров и ячеек памяти до и после выполнения каждой команды, используя отладчик и соответственно команду:

> afdpro hello1.exe.

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.

Протокол 1.

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 DX = 0000 IP = 0010	AX = 1A07 DS = 19F5 DX = 0000 IP = 0013
0013	MOV DS, AX	8ED8	AX = 1A07 DS = 19F5 DX = 0000 IP = 0013	AX = 1A07 DS = 1A07 DX = 0000 IP = 0015
0015	MOV DX, 0000	BA0000	AX = 1A07 DS = 1A07 DX = 0000	AX = 1A07 DS = 1A07 DX = 0000

			IP = 0015	IP = 0018
0018	MOV AH,09	B409	AX = 1A07 DS = 1A07 DX = 0000 IP = 0018	AX = 0907 DS = 1A07 DX = 0000 IP = 001A
001A	INT 21	CD21	AX = 0907 DS = 1A07 DX = 0000 IP = 001A CS = 1A05	AX = 0907 DS = 1A07 DX = 0000 IP = 14A0 CS = F000
001C	MOV AH,4C	B44C	AX = 0907 DS = 1A07 DX = 0000 IP = 001C	AX = 4C07 DS = 1A07 DX = 0000 IP = 001E
001E	INT 21	CD21	AX = 4C07 DS = 1A07 DX = 0000 IP = 001E CS = 1A05 SP = 0100	AX = 0000 DS = 19F5 DX = 0000 IP = 0010 CS = F000 SP = 00F4

Часть 2.

1. Просмотрен код программы hello2.asm, изучены структура и реализация сегментов программы. Строки-приветствия преобразованы в соответствии с личными данными.

2. Выполнена трансляция программы hello2.asm командой:

```
>masm hello2.asm
```

Создан объектный файл HELLO2.obj с сообщением об отсутствии ошибок.

3. Скомпилирован загрузочный модуль HELLO2.exe командой:

>link Hello2.obj

4. Выполнена программа HELLO2.exe в автоматическом режиме и проверено, что она работает корректно: в консоль выводится:

>Hello World!

>Student from 9383 – Orlov Daniil

5. Запущена программа HELLO2.exe в пошаговом режиме, используя отладчик afdpro с фиксацией содержимого используемых регистров и ячеек памяти до и после выполнения команд.

Протокол 2

Содержимое сегментных регистров до старта программы:
CS:1A05, DS:19F5, ES:19F5, SS:1A0A, HS:19F5, FS:19F5

Адрес Команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	До выполнения
0005	PUSH DS	1E	SP = 0018 IP = 0005 DS = 19F5 Stack +0 0000 +2 0000 +4 0000	SP = 0016 IP = 0006 DS = 19F5 Stack +0 19F5 +2 0000 +4 0000
0006	SUB AX, AX	2BC0	AX = 0000 IP = 0006	AX = 0000 IP = 0008
0008	PUSH AX	50	AX = 0000 SP = 0016 IP = 0008 Stack +0 19F5 +2 0000 +4 0000	AX = 0000 SP = 0014 IP = 0009 Stack +0 0000 +2 19F5 +4 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	BA0000	DX = 0000 IP = 000E	DX = 0000 IP = 0011
0011	CALL 0000	E8ECFF	SP = 0014 IP = 0011 Stack +0 0000 +2 19F5 +4 0000	SP = 0012 IP = 0000 Stack +0 0014 +2 0000 +4 19F5
0000	MOV AH,09	B409	AX = 1A07 IP = 0000	AX = 0907 IP = 0002
0002	INT 21	CD21	SP = 0012 IP = 0002 CS = 1A0A Stack +0 0014 +2 0000 +4 19F5 +6 0000	SP = 0006 IP = 14A0 CS = F000 Stack +0 42BD +2 06C5 +4 7044 +6 0004
0004	RET	C3	SP = 0012 IP = 0004 Stack +0 0014 +2 0000 +4 19F5	SP = 0014 IP = 0014 Stack +0 0000 +2 19F5 +4 0000
0014	MOV DX,000F	BA0F00	DX = 0000 IP = 0014	DX = 000F 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 Stack +0 001A +2 0000 +4 19F5	SP = 0014 IP = 001A Stack +0 0000 +2 19F5 +4 0000
001A	RET Far	CB	SP = 0014 IP = 001A CS = 1A0A Stack +0 0000 +2 19F5 +4 0000	SP = 0018 IP = 0000 CS = 19F5 Stack +0 0000 +2 0000 +4 0000
0000	INT 20	CD20	AX = 0907 DX = 0010 DS = 1A07 IP = 0000	AX = 0000 DX = 0000 DS = 19F5 IP = 0005

Выводы.

В результате выполнения лабораторной работы была освоена трансляция, изучено, как происходит выполнение и отладка программ на языке Ассемблера, а также разобраны структуры приведенных в работе программ.

ПРИЛОЖЕНИЕ А. ИСХОДНЫЙ КОД ПРОГРАММЫ

Текст 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 'Вас приветствует ст.гр.9383 - Орлов Д. 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

Текст 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

AStack  SEGMENT STACK
        DW 12 DUP(?) ; Отводится 12 слов памяти
AStack  ENDS

; Данные программы

DATA    SEGMENT

; Директивы описания данных

HELLO    DB 'Hello World!', 0AH, 0DH, EOFLine
GREETING DB 'Student from 9383 - Orlov Daniil$'
DATA     ENDS
```

; Код программы

CODE SEGMENT

ASSUME CS:Code DS:DATA SS:AStack

; 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

Текст файла hello1.lst

Microsoft (R) Macro Assembler Version 5.10

12/14/20 02:57:4

Page 1-1

```
; HELLO1.ASM - упрощенная версия учебной прог  
аммы лаб.раб. N1  
  
; по дисциплине "Архитектура комп  
ьютера"  
  
; *****  
  
*****  
  
; Назначение: Программа формирует и выводит на  
экран приветствие  
  
; пользователя с помощью функции ДО  
С "Вывод строки"  
  
; (номер 09 прерывание 21h), котора  
я:  
  
; - обеспечивает вывод на экран ст  
роки символов,  
  
; заканчивающейся знаком "$";  
  
; - требует задания в регистре ah  
номера функции=09h,  
  
; а в регистре dx - смещения а  
дреса 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

0000 Greeting LABEL BYTE

; Текст приветствия

0000 82 A0 E1 20 AF E0 DB 'Вас приветствует ст.гр.9383 - Орлов Д. С

.,13,10','\$'

A8 A2 A5 E2 E1 E2

A2 E3 A5 E2 20 E1

E2 2E A3 E0 2E 39

33 38 33 20 2D 20

8E E0 AB AE A2 20

84 2E 20 91 2E 0D

0A 24

.CODE ; Начал

о сегмента кода

0000 B8 ---- R mov ax, @data ; Загр

узка в DS адреса начала

0003 8E D8 mov ds, ax ; сегм

ента данных

0005 BA 0000 R mov dx, OFFSET Greeting ; Загр

узка в dx смещения

Microsoft (R) Macro Assembler Version 5.10 12/14/20 02:57:4

Page 1-2

; адрес

а текста приветствия

0008 DisplayGreeting:

0008 B4 09 mov ah, 9 ; # фу

нкции ДОС печати строки

000A CD 21 int 21h ; вывод

на экран приветствия

000C B4 4C mov ah, 4ch ; # фу

нкции ДОС завершения программы

000E CD 21 int 21h ; завер

шение программы и выход в ДОС

END

Microsoft (R) Macro Assembler Version 5.10

12/14/20 02:57:4

Symbols-1

Segments and Groups:

N a m e	Length	Align	Combine	Class
DGROUP	GROUP			
_DATA	002C	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

47992 + 461315 Bytes symbol space free

0 Warning Errors

0 Severe Errors

Текст файла hello2.lst

Microsoft (R) Macro Assembler Version 5.10

12/14/20 02:58:0

Page 1-1

; HELLO2 - Учебная программа N2 лаб.раб.#1 по
дисциплине "Архитектура компьютера"


```

; Программа использует процедуру для п
ечати строки
;
; ТЕКСТ ПРОГРАММЫ

= 0024 EOFLine EQU '$' ; Определение символъ
ной константы
; "Конец строки"

; Стек программы

0000 AStack SEGMENT STACK
0000 000C[ DW 12 DUP(?) ; Отводится 12 слов п
амяти
????
]

0018 AStack ENDS

; Данные программы

0000 DATA SEGMENT

; Директивы описания данных

0000 48 65 6C 6C 6F 20 HELLO DB 'Hello World!', 0AH, 0DH, EOFLine
57 6F 72 6C 64 21
0A 0D 24
000F 53 74 75 64 65 6E GREETING DB 'Student from 9383 - Orlov Daniil$
,

```

74 20 66 72 6F 6D

20 39 33 38 33 20

2D 20 4F 72 6C 6F

76 20 44 61 6E 69

69 6C 24

0030 DATA ENDS

; Код программы

0000 CODE SEGMENT

ASSUME CS:Code DS:DATA SS:AStack

hello2.asm(28): warning A4001: Extra characters on line

; Процедура печати строки

0000 WriteMsg PROC NEAR

0000 B4 09 mov AH,9

0002 CD 21 int 21h ; Вызов функции DOS по пре
рыванию

0004 C3 ret

0005 WriteMsg ENDP

; Головная процедура

0005 Main PROC FAR

0005 1E push DS ;\ Сохранение адреса

Microsoft (R) Macro Assembler Version 5.10 12/14/20 02:58:0

Page 1-2

начала PSP в стеке

0006 2B C0 sub AX,AX ; > для последующего в
осстановления по

```

0008 50                push AX    ;/ команде ret, завер
                          шающей процедуру.
0009 B8 ---- R        mov  AX,DATA    ; Загрузка
                          сегментного
000C 8E D8            mov  DS,AX      ; регистра
                          данных.
000E BA 0000 R        mov  DX, OFFSET HELLO  ; Вывод на
                          экран первой
0011 E8 0000 R        call WriteMsg    ; строки пр
                          иветствия.
0014 BA 000F R        mov  DX, OFFSET GREETING ; Вывод на
                          экран второй
0017 E8 0000 R        call WriteMsg    ; строки пр
                          иветствия.
001A CB              ret              ; Выход в D
                          OS по команде,
                          ; находящей
                          ся в 1-ом слове PSP.
001B                Main    ENDP
001B                CODE    ENDS
                          END Main

```

Microsoft (R) Macro Assembler Version 5.10 12/14/20 02:58:0

Symbols-1

Segments and Groups:

N a m e	Length	Align	Combine	Class
ASTACK	0018	PARA	STACK	

CODE	001B	PARA	NONE
DATA	0030	PARA	NONE

Symbols:

N a m e	Type	Value	Attr
EOFLINE	NUMBER	0024	
GREETING	L BYTE	000F	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	

51 Source Lines

51 Total Lines

13 Symbols

48000 + 461307 Bytes symbol space free

1 Warning Errors

0 Severe Errors