

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: Трансляции, отладка и выполнение программ на языке
Ассемблер

Студент гр. 9383

Крейсманн К.В.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

Цель работы.

Познакомиться с трансляцией, отладкой и выполнением программ на языке Ассемблера.

Задание:

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. 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.

Текст исходного 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 - Крейсманн К.В.',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 исходного 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 Worlds!', 0AH, 0DH,EOFLine

GREETING DB 'Student from 9383 - Kreysmann Kirill \$'

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

Файл диагностических сообщений hello1.lst:

Microsoft (R) Macro Assembler Version 5.10

9/22/20 08:00:28

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 - Крейсманн
К.В.',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

8A E0 A5 A9 E1 AC

A0 AD AD 20 8A 2E

82 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 9/22/20 08:00:28

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

9/22/20 08:00:28

Symbols-1

Segments and Groups:

<i>N a m e</i>	<i>Length</i>	<i>Align</i>	<i>Combine</i>	<i>Class</i>
<i>DGROUP</i>	<i>GROUP</i>			
<i>_DATA</i>	<i>002F</i>	<i>WORDPUBLIC</i>		<i>'DATA'</i>
<i>STACK</i>	<i>0100</i>	<i>PARA STACK</i>		<i>'STACK'</i>
<i>_TEXT</i>	<i>0010</i>	<i>WORDPUBLIC</i>		<i>'CODE'</i>

Symbols:

<i>N a m e</i>	<i>Type</i>	<i>Value</i>	<i>Attr</i>
<i>DISPLAYGREETING</i>	<i>L NEAR</i>	<i>0008</i>	<i>_TEXT</i>
<i>GREETING</i>	<i>L BYTE</i>	<i>0000</i>	<i>_DATA</i>
<i>@CODE</i>	<i>TEXT</i>	<i>_TEXT</i>	
<i>@CODESIZE</i>	<i>TEXT</i>	<i>0</i>	
<i>@CPU</i>	<i>TEXT</i>	<i>0101h</i>	
<i>@DATASIZE</i>	<i>TEXT</i>	<i>0</i>	
<i>@FILENAME</i>	<i>TEXT</i>	<i>hello1</i>	
<i>@VERSION</i>	<i>TEXT</i>	<i>510</i>	

33 Source Lines

33 Total Lines

19 Symbols

47994 + 461313 Bytes symbol space free

0 Warning Errors

0 Severe Errors

Файл диагностических сообщений hello2.lst:

Microsoft (R) Macro Assembler Version 5.10

9/22/20 08:10:34

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 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 9383 - Kreysmann Kir
ill \$'

74 20 66 72 6F 6D

20 39 33 38 33 20

```

2D 20 4B 72 65 79
73 6D 61 6E 6E 20
4B 69 72 69 6C 6C
20 24
0036          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
Microsoft (R) Macro Assembler Version 5.10          9/22/20 08:10:34
               Page    1-2
0005 1E              push DS      ;\ Сохранение адреса
               начала PSP в стеке
0006 2B C0             sub  AX,AX  ; > для последующего в
               останова 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 0010 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 9/22/20 08:10:34

Symbols-1

Segments and Groups:

<i>N a m e</i>	<i>Length</i>	<i>Align</i>	<i>Combine</i>	<i>Class</i>
ASTACK.....	0018	PARA	STACK	
CODE.....	001B	PARA	NONE	
DATA.....	0036	PARA	NONE	

Symbols:

<i>N a m e</i>	<i>Type</i>	<i>Value</i>	<i>Attr</i>
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	

51 Source Lines

51 Total Lines

13 Symbols

48002 + 461305 Bytes symbol space free

1 Warning Errors

0 Severe Errors

Выполнение работы:

1. Просмотрел файлы hello1.asm и hello2.asm.
2. Разобрался в структуре и реализации программы. Преобразовал строку приветствия в обоих файлах в соответствии со своими личными данными.
3. Протранслировал файлы hello1.asm и hello2.asm с помощью строк

>masm hello1.asm

>masm hello2.asm

В файле hello2.asm была обнаружена ошибка

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

28 строку преобразовал в следующий вид :

ASSUME CS:Code, DS:DATA, SS:AStack

4. Скомпоновал загрузочные модули с помощью ввода строк:

>link hello1.obj

>link hello2.obj

5. Запустил программы в автоматическом режиме путем набора строк:

>hello1.exe

Результат: «Вас приветствует ст.гр.9383 – Крейсманн К.В.»

>hello2.exe

Результат: «Hello worlds!

Student from 9383 – Kreysmann Kirill»

6. Запустил выполнение программ под управлением отладчика с помощью команд:

>afdp Hello1.exe

>afdp Hello2.exe

Протокол пошагового исполнения программы 1:

Начальные значения: (CS) = 1A05, (DS) = 19F5, (ES) = 19F5, (SS) = 1A0A

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти до	Содержимое регистров и ячеек памяти после
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,9h	B409	(AX) = 1A07 (IP) = 0018	(AX) = 0907 (IP) = 001A
001A	Int 21h	CD21	(IP) = 001A	(IP) = 001C
001C	Mov AH,4C	B44C	(AX) = 0907 (IP) = 001C	(AX) = 4C07 (IP) = 001E
001E	Int 21h	CD21	(IP) = 001E	(IP) = 0010

Протокол пошагового исполнения программы 2:

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

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти до	Содержимое регистров и ячеек памяти после
0005	Push DS	1E	(DS) = 19F5 Stack +0 0000 (SP) = 0018 +2 0000 (IP) = 0005 +4 0000 +6 0000	(DS) = 19F5 Stack +0 19F5 (SP) = 0016 +2 0000 (IP) = 0006 +4 0000 +6 0000
0006	Sub AX,AX	2BC0	(AX) = 0000 (IP) = 0006	(AX) = 0000 (IP) = 0008
0008	PUSH AX	50	(AX) = 0000 Stack +0 19F5 (SP) = 0016 +2 0000 (IP) = 0008 +4 0000 +6 0000	(AX) = 0000 Stack +0 0000 (SP) = 0014 +2 19F5 (IP) = 0009 +4 0000 +6 0000
0009	MOV AX,1A07	B8071A	(AX) = 0000 (IP) = 0009	(AX) = 1A07 (IP) = 000C
000C	MOV DS,AX	BED8	(DS) = 19F5 (AX) = 1A07 (IP) = 000C	(DS) = 1A07 (AX) = 1A07 (IP) = 000E
000E	MOV DX, 0000	BA0000	(DX) = 0000 (IP) = 000E	(DX) = 0000 (IP) = 0011
0011	Call 0000	E8ECFF	(SP) = 0014 Stack +0 0000 (IP) = 0011 +2 19F5 +4 0000 +6 0000	(SP) = 0012 Stack +0 0014 (IP) = 0000 +2 0000 +4 19F5 +6 0000
0000	Mov AH,9h	B409	(AX) = 1A07 (IP) = 0000	(AX) = 0907 (IP) = 0002
0002	Int 21h	CD21	(IP) = 0002	(IP) = 0004
0004	Ret	C3	(IP) = 0004 Stack +0 0014 (SP) = 0012 +2 0000 +4 19F5 +6 0000	(IP) = 0014 Stack +0 0000 (SP) = 0014 +2 19F5 +4 0000 +6 0000
0014	Mov DX, 0010	BA1000	(DX) = 0000 (IP) = 0014	(DX) = 0010 (IP) = 0017
0017	Call 0000	E6FF	(SP) = 0014 Stack +0 0000 (IP) = 0017 +2 19F5 +4 0000 +6 0000	(SP) = 0012 Stack: +0 001A (IP) = 0000 +2 0000 +4 19F5 +6 0000
0000	Mov AH,9h	B409	(AX) = 0907 (IP) = 0000	(AX) = 0907 (IP) = 0002
0002	Int 21h	CD21	(IP) = 0002	(IP) = 0004
0004	Ret	C3	(IP) = 0004 Stack +0 001A (SP) = 0012 +2 0000 +4 19F5 +6 0000	(IP) = 001A Stack +0 0000 (SP) = 0014 +2 19F5 +4 0000 +6 0000

001A	Ret Far	CB	(IP) = 001A	(IP) = 0000 (CS) = 19F5
0000	Int 20	CD20	(IP) = 0000	(IP) = 0005

Вывод:

Было произведено знакомство с трансляцией, отладкой и выполнением программ на языке Ассемблера. Получено базовое представление о программах на языке Ассемблер.