

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: Трансляции, отладка и выполнение программ на языке
Ассемблера.

Студент гр. 9383

Ноздрин В.Я.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

Цель работы.

Изучить основные принципы трансляции, отладки и выполнения программ на языке Ассемблера.

Ход работы.

Часть 1.

1. Загружены файлы из каталога /tools в каталог ~/tools
2. Запущена программа DOSBox, смонтирован виртуальный диск C: в каталоге ~/tools при помощи mount c ~/tools
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 c:
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 — Ноздрин В. Я.
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.

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 .	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 (IP) = 0015	(AX) = 1A07 (DS) = 1A07 (DX) = 0000 (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	(AX) = 0907 (DS) = 1A07 (DX) = 0000 (IP) = 001C
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	(AX) = 0000 (DS) = 19F5 (DX) = 0000 (IP) = 0010

Табл.1

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 Worlds!

>Student from 9383 - Nozdrin V.Y..

5. Запущена программа Hello2.exe в пошаговом режиме, используя отладчик afdpro с фиксацией содержимого используемых регистров и ячеек памяти до и после выполнения команд.

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

Адрес Команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0005	PUSH DS	1E	(AX) = 0000 (DS) = 19F5 (SP) = 0018 (IP) = 0005 Stack +0 0000 +2 0000 +4 0000	(AX) = 0000 (DS) = 19F5 (SP) = 0016 (IP) = 0006 Stack +0 19F5 +2 0000 +4 0000
0006	SUB AX, AX	2BC0	(AX) = 0000 (DS) = 19F5 (SP) = 0016 (IP) = 0006 Stack +0 19F5 +2 0000 +4 0000	(AX) = 0000 (DS) = 19F5 (SP) = 0016 (IP) = 0008 Stack +0 19F5 +2 0000 +4 0000
0008	PUSH AX	50	(AX) = 0000 (DS) = 19F5 (SP) = 0016 (IP) = 0008 Stack +0 19F5 +2 0000 +4 0000	(AX) = 0000 (DS) = 19F5 (SP) = 0014 (IP) = 0009 Stack +0 0000 +2 19F5 +4 0000
0009	MOV AX, 1A07	B8071A	(AX) = 0000 (DS) = 19F5 (SP) = 0014 (IP) = 0009 Stack +0 0000 +2 19F5 +4 0000	(AX) = 1A07 (DS) = 19F5 (SP) = 0014 (IP) = 000C Stack +0 0000 +2 19F5 +4 0000

000C	MOV DS, AX	8ED8	(AX) = 1A07 (DS) = 19F5 (SP) = 0014 (IP) = 000C Stack +0 0000 +2 19F5 +4 0000	(AX) = 1A07 (DS) = 1A07 (SP) = 0014 (IP) = 000E Stack +0 0000 +2 19F5 +4 0000
000E	MOV DX, 0000	BA0000	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (SP) = 0014 (IP) = 000E Stack +0 0000 +2 19F5 +4 0000	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (SP) = 0014 (IP) = 0011 Stack +0 0000 +2 19F5 +4 0000
0011	CALL 0000	E8ECFF	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (SP) = 0014 (IP) = 0011 Stack +0 0000 +2 19F5 +4 0000	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (SP) = 0012 (IP) = 0000 Stack +0 0014 +2 0000 +4 19F5
0000	MOV AH,09	B409	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (SP) = 0012 (IP) = 0000 Stack +0 0014 +2 0000 +4 19F5	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (SP) = 0012 (IP) = 0002 Stack +0 0014 +2 0000 +4 19F5
0002	INT 21	CD21	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (SP) = 0012 (IP) = 0002 Stack +0 0014 +2 0000 +4 19F5	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (SP) = 0012 (IP) = 0004 Stack +0 0014 +2 0000 +4 19F5
0004	RET	C3	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (SP) = 0012 (IP) = 0004 Stack +0 0014 +2 0000 +4 19F5	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (SP) = 0014 (IP) = 0014 Stack +0 0000 +2 19F5 +4 0000

0014	MOV DX,0010	BA1000	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (SP) = 0014 (IP) = 0014 Stack +0 0000 +2 19F5 +4 0000	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (SP) = 0014 (IP) = 0017 Stack +0 0000 +2 19F5 +4 0000
0017	CALL 0000	E8E6FF	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (SP) = 0014 (IP) = 0017 Stack +0 0000 +2 19F5 +4 0000	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (SP) = 0012 (IP) = 0000 Stack +0 001A +2 0000 +4 19F5
0000	MOV AH,09	B409	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (SP) = 0012 (IP) = 0000 Stack +0 001A +2 0000 +4 19F5	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (SP) = 0012 (IP) = 0002 Stack +0 001A +2 0000 +4 19F5
0002	INT 21	CD21	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (SP) = 0012 (IP) = 0002 Stack +0 001A +2 0000 +4 19F5	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (SP) = 0012 (IP) = 0004 Stack +0 001A +2 0000 +4 19F5
0004	RET	C3	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (SP) = 0012 (IP) = 0004 Stack +0 001A +2 0000 +4 19F5	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (SP) = 0014 (IP) = 001A Stack +0 0000 +2 19F5 +4 0000
001A	RET Far	CB	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (SP) = 0014 (IP) = 001A Stack +0 0000 +2 19F5 +4 0000	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (SP) = 0018 (IP) = 0000 Stack +0 0000 +2 0000 +4 0000

0000	INT 20	CD20	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (SP) = 0018 (IP) = 0000 Stack +0 0000 +2 0000 +4 0000	(AX) = 0000 (DX) = 0000 (DS) = 19F5 (SP) = 0018 (IP) = 0005 Stack +0 0000 +2 0000 +4 0000
------	--------	------	--	--

Табл.2.

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 - Ноздрин В.Я.',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 hello1.lst

Microsoft (R) Macro Assembler Version 5.10

10/15/20 09:59:1

Page 1-1

```
; 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 ДОО "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 ДОО

.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 - Ноздрин В.
Я.',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

8D AE A7 A4 E0 A8

AD 20 82 2E 9F 2E

0D 0A 24

.CODE

; Начало сегмента кода

0000 B8 ---- R

mov ax, @data

; Загрузка в DS адреса

начала

0003 8E D8

mov ds, ax

; сегмента данных

0005 BA 0000 R

mov dx, OFFSET Greeting

; Загрузка в dx

смещения

; адреса текста приветствия

```

0008          DisplayGreeting:
0008 B4 09          mov ah, 9          ; # функции ДОС печати
строки
000A CD 21          int 21h          ; вывод на экран
приветствия
000C B4 4C          mov ah, 4ch      ; # функции      ДОС
завершения программы
000E CD 21          int 21h          ; завершение программы и
выход в ДОС

```

END

Segments and Groups:

Name	Length	Align	Combine	Class
DGROUP	GROUP			
_DATA	002D	WORD	PUBLIC	'DATA'
_STACK	0100	PARA	STACK	'STACK'
_TEXT	0010	WORD	PUBLIC	'CODE'

Symbols:

Name	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	

32 Source Lines

32 Total Lines

19 Symbols

47994 + 461313 Bytes symbol space free

0 Warning Errors

0 Severe Errors

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 '$'      ; Определение символьной константы
                        ; "Конец строки"
; Стек программы
AStack SEGMENT STACK
        DW 12 DUP(?)  ; Отводится 12 слов памяти
AStack ENDS
; Данные программы
DATA SEGMENT
; Директивы описания данных
HELLO DB 'Hello Worlds!', 0AH, 0DH, EOFLine
GREETING DB 'Student from 9383 - Nozdrin V.Y.$'
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 hello2.asm

Microsoft (R) Macro Assembler Version 5.10

10/15/20 11:07:2

Page 1-1

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

```

; Программа использует процедуру для печати строки
;
; ТЕКСТ ПРОГРАММЫ

= 0024                                EOFLine EQU '$'           ; Определение
символьной константы

; "Конец строки"
; Стек программы

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

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 - Nozdrin V.Y.$'
74 20 66 72 6F 6D
20 39 33 38 33 20
2D 20 4E 6F 7A 64
72 69 6E 20 56 2E
59 2E 24
0031                                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 прерыванию
0004 C3                                ret
0005                                WriteMsg ENDP
; Головная процедура

0005                                Main PROC FAR
0005 1E                                push DS ;\ Сохранение адреса

```

начала PSP в стеке

```

0006 2B C0          sub    AX,AX      ; > для последующего
восстановления по
0008 50             push   AX        ;/ команде ret, завершающей
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 приветствия.
0014 BA 0010 R      mov    DX, OFFSET GREETING ; Вывод на экран
второй
0017 E8 0000 R      call   WriteMsg    ; строки приветствия.
001A CB             ret              ; Выход в DOS по команде,
; находящейся в 1-ом слове PSP.

001B               Main    ENDP
001B               CODE    ENDS
END Main

```

Segments and Groups:

Name	Length	Align	Combine	Class
ASTACK.....	0018	PARA	STACK	
CODE.....	001B	PARA	NONE	
DATA.....	0031	PARA	NONE	

Symbols:

Name	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	

51 Source Lines

51 Total Lines

13 Symbols

48002 + 461305 Bytes symbol space free

1 Warning Errors

0 Severe Errors