

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.

Студентка гр. 9383

Карпекина А.А.

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. Перемещены файлы из папки tools в каталог C:/LAB.
2. Запущена программа DOSBox, смонтирован виртуальный диск C: в каталоге /LAB с помощью команды mount c c:/LAB .
3. Произведен просмотр кода с помощью редактора Atom, для смены кодировки и изменения выводимых данных.
4. В DOS осуществлен переход на виртуальный диск при помощи команды C:\.
5. Протранслирована программа с помощью команды:
masm HELLO1.ASM
В процессе создается объектный файл HELLO1.OBJ и файл диагностических сообщений HELLO1.LST.
6. Скомпонован загрузочный модуль HELLO1.EXE с помощью команды:
link HELLO1.OBJ
В результате работы линковщика создается загрузочный модуль HELLO1.EXE.
7. Для загрузки русских символов вводим команду:
keyb ru 866
8. Запуск программы:
HELLO1.EXE
9. Вывод программы:
Вас приветствует ст.гр.9383 — Карпекина А.А.

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: `afdp HELLO1.EXE`.

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

Таблица 1 - Результаты работы

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

Часть 2.

1. Протранслирована программа с помощью команды:
`masm HELLO2.ASM`

В процессе создается объектный файл HELLO2.OBJ и файл диагностических сообщений HELLO2.LST.

2. Скомпонован загрузочный модуль HELLO2.EXE с помощью команды:
link HELLO2.OBJ

В результате работы линковщика создается загрузочный модуль HELLO2.EXE.

7. Для загрузки русских символов вводим команду:
keyb ru 866

8. Запуск программы:
HELLO2.EXE

9. Вывод программы:
Вас приветствует ст.гр.9383 — Карпекина А.А.

10. Выполнен запуск программы HELLO2.EXE в пошаговом режиме с фиксацией используемых регистров и ячеек памяти до и после выполнения каждой команды, используя отладчик и соответственно команду:
afopro HELLO2.EXE.

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

Таблица 2 - результаты работы

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 (CS) = 1A0A Stack +0 0000 +2 0000 +4 0000	(AX) = 0000 (DS) = 19F5 (SP) = 0016 (IP) = 0006 (CS) = 1A0A Stack +0 19F5 +2 0000 +4 0000
0006	SUB AX, AX	2BC0	(AX) = 0000 (DS) = 19F5 (SP) = 0016 (IP) = 0006 (CS) = 1A0A Stack +0 19F5 +2 0000 +4 0000	(AX) = 0000 (DS) = 19F5 (SP) = 0016 (IP) = 0008 (CS) = 1A0A Stack +0 19F5 +2 0000 +4 0000

Таблица 2 - Продолжение

0008	PUSH AX	50	(AX) = 0000 (DS) = 19F5 (SP) = 0016 (IP) = 0008 (CS) = 1A0A Stack +0 19F5 +2 0000 +4 0000	(AX) = 0000 (DS) = 19F5 (SP) = 0014 (IP) = 0009 (CS) = 1A0A Stack +0 0000 +2 19F5 +4 0000
0009	MOV AX, 1A07	B8071A	(AX) = 0000 (DS) = 19F5 (SP) = 0014 (IP) = 0009 (CS) = 1A0A Stack +0 0000 +2 19F5 +4 0000	(AX) = 1A07 (DS) = 19F5 (SP) = 0014 (IP) = 000C (CS) = 1A0A Stack +0 0000 +2 19F5 +4 0000
000C	MOV DS, AX	8ED8	(AX) = 1A07 (DS) = 19F5 (SP) = 0014 (IP) = 000C (CS) = 1A0A Stack +0 0000 +2 19F5 +4 0000	(AX) = 1A07 (DS) = 1A07 (SP) = 0014 (IP) = 000E (CS) = 1A0A Stack +0 0000 +2 19F5 +4 0000
000E	MOV DX, 0000	BA0000	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (SP) = 0014 (IP) = 000E (CS) = 1A0A Stack +0 0000 +2 19F5 +4 0000	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (SP) = 0014 (IP) = 0011 (CS) = 1A0A Stack +0 0000 +2 19F5 +4 0000
0011	CALL 0000	E8ECFF	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (SP) = 0014 (IP) = 0011 (CS) = 1A0A Stack +0 0000 +2 19F5 +4 0000	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (SP) = 0012 (IP) = 0000 (CS) = 1A0A Stack +0 0014 +2 0000 +4 19F5

Таблица 2 - Продолжение

0000	MOV AH,09	B409	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (SP) = 0012 (IP) = 0000 (CS) = 1A0A Stack +0 0014 +2 0000 +4 19F5	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (SP) = 0012 (IP) = 0002 (CS) = 1A0A Stack +0 0014 +2 0000 +4 19F5
0002	INT 21	CD21	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (SP) = 0012 (IP) = 0002 (CS) = 1A0A Stack +0 0014 +2 0000 +4 19F5	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (SP) = 0012 (IP) = 0004 (CS) = 1A0A Stack +0 0014 +2 0000 +4 19F5
0004	RET	C3	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (SP) = 0012 (IP) = 0004 (CS) = 1A0A Stack +0 0014 +2 0000 +4 19F5	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (SP) = 0014 (IP) = 0014 (CS) = 1A0A Stack +0 0000 +2 19F5 +4 0000
0014	MOV DX,0010	BA1000	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (SP) = 0014 (IP) = 0014 (CS) = 1A0A Stack +0 0000 +2 19F5 +4 0000	(AX) = 0907 (DX) = 0006 (DS) = 1A07 (SP) = 0014 (IP) = 0017 (CS) = 1A0A Stack +0 0000 +2 19F5 +4 0000
0017	CALL 0000	E8E6FF	(AX) = 0907 (DX) = 0006 (DS) = 1A07 (SP) = 0014 (IP) = 0017 (CS) = 1A0A Stack +0 0000 +2 19F5 +4 0000	(AX) = 0907 (DX) = 0006 (DS) = 1A07 (SP) = 0012 (IP) = 0000 (CS) = 1A0A Stack +0 001A +2 0000 +4 19F5

Таблица 2 - Продолжение

0000	MOV AH,09	B409	(AX) = 0907 (DX) = 0006 (DS) = 1A07 (SP) = 0012 (IP) = 0000 (CS) = 1A0A Stack +0 001A +2 0000 +4 19F5	(AX) = 0907 (DX) = 0006 (DS) = 1A07 (SP) = 0012 (IP) = 0002 (CS) = 1A0A Stack +0 001A +2 0000 +4 19F5
0002	INT 21	CD21	(AX) = 0907 (DX) = 0006 (DS) = 1A07 (SP) = 0012 (IP) = 0002 (CS) = 1A0A Stack +0 001A +2 0000 +4 19F5	(AX) = 0907 (DX) = 0006 (DS) = 1A07 (SP) = 0012 (IP) = 0004 (CS) = 1A0A Stack +0 001A +2 0000 +4 19F5
0004	RET	C3	(AX) = 0907 (DX) = 0006 (DS) = 1A07 (SP) = 0012 (IP) = 0004 (CS) = 1A0A Stack +0 001A +2 0000 +4 19F5	(AX) = 0907 (DX) = 0006 (DS) = 1A07 (SP) = 0014 (IP) = 001A (CS) = 1A0A Stack +0 0000 +2 19F5 +4 0000
001A	RET Far	CB	(AX) = 0907 (DX) = 0006 (DS) = 1A07 (SP) = 0014 (IP) = 001A (CS) = 1A0A Stack +0 0000 +2 19F5 +4 0000	(AX) = 0907 (DX) = 0006 (DS) = 1A07 (SP) = 0018 (IP) = 0000 (CS) = 19F5 Stack +0 0000 +2 0000 +4 0000
0000	INT 20	CD20	(AX) = 0907 (DX) = 0006 (DS) = 1A07 (SP) = 0018 (IP) = 0000 (CS) = 19F5 Stack +0 0000 +2 0000 +4 0000	(AX) = 0000 (DX) = 0000 (DS) = 19F5 (SP) = 0018 (IP) = 0005 (CS) = 19F5 Stack +0 0000 +2 0000 +4 0000

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

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 - Карпекина А.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

```

Текст файла hello1.lst

```

; HELLO1.ASM - упрощенная версия
учебной прогр
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
- Карпекина
      A.A.',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 A0 E0 AF A5 AA
      A8 AD A0 20 80 2E
      80 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
9/30/20 21:46:13

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
; # фy
нкции ДОС завершения программы
000E CD 21 int 21h
; завер
шение программы и выход в ДОС
END

```

Microsoft (R) Macro Assembler Version 5.10
9/30/20 21:46:13

Symbols-1

Segments and Groups:

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

47994 + 461313 Bytes symbol space free

0 Warning Errors

0 Severe Errors

ПРИЛОЖЕНИЕ В

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 'Hi!', 0AH, 0DH, EOFLine
GREETING DB 'Student from 9383 - Karpekina A.A. \$'
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.lst

```

                                ; 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 69 21 0A 0D 24 HELLO      DB 'Hi!', 0AH,
0DH,EOFLine
0006  53 74 75 64 65 6E GREETING  DB 'Student from 9383 -
Karpekina A.A
        . $'
        74 20 66 72 6F 6D
        20 39 33 38 33 20
        2D 20 4B 61 72 70
        65 6B 69 6E 61 20
        41 2E 41 2E 20 24
002A          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
начала PSP в стеке
0006  2B C0                                sub  AX,AX ; > для
последующего в
Microsoft (R) Macro Assembler Version 5.10
9/30/20 23:06:10

```

```

                                остановления 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 0006 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
9/30/20 23:06:10

```

Symbols-1

Segments and Groups:

	N a m e	Length	Align
Combine Class			
ASTACK	0018	PARA STACK
CODE	001B	PARA NONE
DATA	002A	PARA NONE

Symbols:

	N a m e	Type	Value	Attr
	EOFLINE	NUMBER	0024	
	GREETING	L BYTE	0006	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