

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: Трансляция, отладка и выполнение программ на языке Ассемблера

Студентка гр. 1381

Деркачева Дарья

Преподаватель

Ефремов М. А.

Санкт-Петербург

2022

Цель работы

Научиться транслировать (транслятор `masm`), компоновать (компоновщик `link`), отлаживать (отладчик `afd`) и выполнять простые программы на ассемблере при помощи DOSBOX.

Текст задания

Выполнение работы состоит из двух частей, по каждой из которых необходимо представить протокол с фиксацией всех выполняемых действий и полученных результатов.

Часть 1

1. Просмотреть программу `hello1.asm`.
2. Разобраться в структуре и реализации каждого сегмента программы. Непонятные фрагменты прояснить у преподавателя.

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).

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

Выполнить программу в пошаговом режиме с фиксацией используемых регистров и ячеек памяти до и после выполнения каждой команды.

Часть 2

Выполнить пункты 1 - 7 части 1 настоящего задания применительно к программе hello1.asm и hello2.asm.

Текст исходных файлов программ hello1 и hello2

Текст исходной программы hello1 см. в приложении А.

Текст исходной программы hello2 см. в приложении В.

Тексты файлов диагностических hello1.lst и hello2.lst

Текст диагностического файла hello1.lst см. в приложении С

Текст диагностического файла hello2.lst см. в приложении D

Протокол работы на компьютере

1. Загрузить файл hello1.asm, после hello2.asm.

2. Протранслировать каждую из программ с помощью строки

> masm hello1.asm (hello2.asm)

с созданием объектного файла и файла диагностических сообщений (файла листинга).

3. Скомпоновать загрузочный модуль с помощью строки

> link hello1.obj (hello2.obj)

с созданием карты памяти и исполняемого файла hello1.exe, hello2.exe.

4. Выполнить программу в автоматическом режиме путем набора строки

> hello1.exe (или же) hello2.exe

5. Запустить выполнение программы под управлением отладчика с помощью команды

> afd hello1.exe (hello2.exe)

Протокол пошагового исполнения программ

Табл.1. Протокол пошагового исполнения программы hello1

Адрес Команды	Символический код команды	16-ричный команды	Содержимое регистров и ячеек памяти	
			до выполнения	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	(IP) = 0015 (DX)=0000	(IP) = 0018 (DX)=0000
0018	Mov Ah, 09	B409	(IP) = 0018 (AX) = 1A07	(IP) = 001A (AX) = 0907
001A	INT 21	CD21	(IP) = 001A	(IP) = 001C
001C	Mov Ah, 4C	B44C	(IP) = 001C (AX) = 0907	(IP) = 001E (AX) = 4C07
001E	INT 21	CD21	(IP) = 001E	(IP) = 0010
	Program terminated OK			

Табл.1. Протокол пошагового исполнения программы hello2

Адрес Команды	Символический код команды	16-ричный команды	Содержимое регистров и ячеек памяти	
			до выполнения	после выполнения
0005	Push DS	1E	(DS) = 19F5 (SP) = 0018 (IP) = 0005 (Stack) +0 0000	(DS) = 19F5 (SP) = 0016 (IP) = 0006 (Stack) +0 19F5
0006	Sub AX, AX	2BC0	(AX) = 0000	(AX) = 0000

			(IP) = 0006	(IP) = 0008
0008	Push AX	50	(AX) = 0000 (SP) = 0016 (IP) = 0008 (Stack) +0 19F5 (Stack) +2 0000	(AX) = 0000 (SP) = 0014 (IP) = 0009 (Stack) +0 0000 (Stack) +2 19F5
0009	Mov AX, 1A07	B8071A	(AX) = 0000 (IP) = 0009	(AX) = 1A07 (IP) = 000C
000C	Mov DS, AX	8ED8	(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 (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, 09	B409	(AX) = 1A07 (IP) = 0000	(AX) = 0907 (IP) = 0002
0002	INT 21	CD21	(IP) = 0002	(IP) = 0004
0004	RET	C3	(SP) = 0012 (IP) = 0004 (Stack) +0 0014	(SP) = 0014 (IP) = 0014 (Stack) +0 0000

			(Stack) +2 0000 (Stack) +4 19F5	(Stack) +2 19F5 (Stack) +4 0000
0014	Mov DX, 0010	BA1000	(DX) = 0000 (IP) = 0014	(DX) = 0010 (IP) = 0017
0017	Call 0000	E8E6FF	(SP) = 0014 (IP) = 0017 (Stack) +0 0000 (Stack) +2 19F5 (Stack) +4 0000	(SP) = 0012 (IP) = 0000 (Stack) +0 001A (Stack) +2 0000 (Stack) +4 19F5
0000	Mov AH, 09	B409	(IP) = 0000 (AX) = 0907	(IP) = 0002 (AX) = 0907
0002	INT 21	CD21	(IP) = 0002	(IP) = 0004
0004	RET	C3	(SP) = 0012 (IP) = 0004 (Stack) +0 001A (Stack) +2 0000 (Stack) +4 19F5	(SP) = 0014 (IP) = 001A (Stack) +0 0000 (Stack) +2 19F5 (Stack) +4 0000
001A	RET Far	CB	(SP) = 0014 (IP) = 001A (CS) = 1A0A (Stack) +0 0000 (Stack) +2 19F5	(SP) = 0018 (IP) = 0000 (CS) = 19F5 (Stack) +0 0000 (Stack) +2 0000
0000	INT 20	CD20	(IP) = 0000 (AX) = 0907	(IP) = 0005 (AX) = 0000

			(DX) = 0010 (CS) = 19F5	(DX) = 0000 (CS) = 1A0A
	Program terminated OK			

Выводы по работе

В ходе выполнения лабораторной были изучены правила работы с дос, а также синтаксис языка. Появились знания о трансляции, отладке и выполнении программ на языке ассемблера

Приложение А

Текст исходного файла hello1

```

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 студент 1381 Деркачева
Д.',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

```

Приложение В

Текст исходного файла hello2

```

EOFLine    EQU    '$'          ; Определение символьной
константы

                                   ; "Конец строки"

; стек программы

```

```
ASSUME CS:CODE, SS:AStack
```

```
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 1381 - $'
```

```
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,

                                           ; находящейся в 1-ом
слове PSP.

Main     ENDP

CODE     ENDS

        END Main

```

Приложение С

Текст диагностического файла hello1

```

Microsoft (R) Macro Assembler Version 5.10
9/13/22 13:31:37

```

Page 1-1

1

2

```

; HELLO1.ASM -
C:\P\B\P\C%P\PSPSP°C\P I
P\B\P\C\°P C\°P\°P\°P\°P
jPjC< P»P°P±.C°P°P±. N1

```

³ ; PiPs
PrPëCÍC†PëPiP»PëPS
PëPsPjPiCЃCЃ Pµ "PђCЃC...PëC, PµPëC, CÍCЃP°
C, PµCЃP° "

⁴ ;

⁵ ; PќP°P·PSP°C†PµPSPëPµ:
PµCЃPsPiCЃP°PjP
jP° C„PsCЃPjPëCЃCÍPµC, Pë
PIC<PIPsPrPëC
, PSP° CќPëCЃP°PS
PiCЃPëPIPµC, CÍC, PIPëP
µ
⁶ ;
PiPsP»CЃP·PsPIP°C, PµP»CЃ
CÍ PiPsPjPsC%CЃCЃ
C„CÍPSPëC†PëPë P"PhPŸ
"P'C<PIPsPr CÍC, CЃPsPëPë"
⁷ ; (PSPsPjPµCЃ 09
PiCЃPµCЃC<
PIP°PSPëPµ 21h),
PëPsC, PsCЃP°CЃ:
⁸ ; -
PsP†PµCÍPiPµC†PëPIP°Pµ
C, PIC<PIPsPr PSP° CќPëCЃP°PS
CÍC, CЃPsP
ëPë CÍPëPjPIPsP»PsPI,

9
P · P° P e P° P S C † P e P I P° C T h C %

;

P μ P N° C í C ¨ P · P S P° P e P s P j " \$ " ;

10
P · P° P r P

;

- C , C T P μ P ± C í P μ C ,

P S P s P j P μ

° P S P e C ¨ P I C T P μ P i P e C í C , C T P μ a h

C T P° C „ C í P S P e C † P e P e = 0 9 h ,

11
C T P μ P i P e C í C , C T P μ

;

P° P I

P° P r C T P μ C í P°

d x - C í P j P μ C % P μ P S P e C ¨

P I C < P I P s P r P e P j P s P N°

12

;

C í C , C T P s P e P e ;

13

;

-

P e C í P i P s P » C T P · C í P μ C , C

T P μ P i P e C í C , C T a x P e P S P μ

C í P s C ... C T P° P

S C ¨ P μ C , P μ P i P s

14

;

C í P s P r P μ C T P ¨ P e P j P s P μ .

15

;

16

17

DOSSEG

; P - P° P r P° P S P e P μ

C í P μ P i P j P μ P S C ,

P s P I P i P s P r P " P h P Ÿ

```

18                                .MODEL    SMALL
                                   ;    PЃPSPГPμP»CЃ
PіP°PjCЏC,Pё-SMA

                                   LL(PЃP°P»P°CЏ)

19                                .STACK    100h
                                   ;    PЃC,PіPμCЏC,Pё PіPSPГ
PŸC,PμP

                                   є 256 P±P°PN°C,

20                                .DATA
                                   ;    PЃP°C‡P°P»Ps
CЏPμPіPjPμPSC,P°

                                   PГP°PSPSC<C...

21 0000                          Greeting LABEL BYTE
                                   ;    PŸPμPєCЏC,
PіCЃPёPіPμC,CЏC,P

                                   IPёCЏ

22 0000  D0 92 D0 B0 D1 81        DB    'P'P°CЏ
PіCЃPёPіPμC,CЏC,PіCЏPμC,

                                   CЏC,.PіCЃ.7303 - P~PіP°PSPsPI
P~.P~.',1

Microsoft (R) Macro Assembler Version 5.10
9/13/22 13:31:37

Page      1-2

                                   3,10,'$'

23      20 D0 BF D1 80 D0

```

```

24      B8 D0 B2 D0 B5 D1
25      82 D1 81 D1 82 D0
26      B2 D1 83 D0 B5 D1
27      82 20 D1 81 D1 82
28      2E D0 B3 D1 80 2E
29      37 33 30 33 20 2D
30      20 D0 98 D0 B2 D0
31      B0 D0 BD D0 BE D0
32      B2 20 D0 98 2E D0
33      98 2E 0D 0A 24
34      .CODE

```

```

;                                PkP°CþP°P»Ps
CfPµPiPjPµPSC,P° PePsPr

```

P°

```

35 0000 B8 ---- R      mov ax, @data

```

```

; P-P°PiCᵼCfP·PeP° PI DS
P°PrCᵼPµCfP°

```

PSP°CþP°P»P°

```

36 0003 8E D8      mov ds, ax

```

```

;      CfPµPiPjPµPSC,P°
PrP°PSPSC<C...

```

```

37 0005 BA 0000 R      mov dx, OFFSET
Greeting

```

```

; P-P°PiCᵼCfP·PeP° PI dx
CfPjPµC%PµPS

```

PëCᵼ

38


```

; P°PrCṪPμCÍP° C,PμPeCÍC,P°
PìCṪPëPIPμ

C,CÍC,PIPëCṪ

39 0008 DisplayGreeting:
40 0008 B4 09 mov ah, 9
; # C,,CíPSPeC†PëPë P"PḥPŸ
PìPμC†P°C,P

ë CÍC,CṪPsPePë

41 000A CD 21 int 21h
; PIC<PIPsPr PSP° CÍPeCṪP°PS
PìCṪPëPI

PμC,CÍC,PIPëCṪ

42 000C B4 4C mov ah, 4ch
; # C,,CíPSPeC†PëPë P"PḥPŸ
P·P°PIPμCṪC

€PμPSPëCṪ PìCṪPsPìCṪP°PjPjC<

43 000E CD 21 int 21h
; P·P°PIPμCṪC€PμPSPëPμ
PìCṪPsPìCṪP°PjP

jC< Pë PIC<C...PsPr PI P"PḥPŸ

44 END

```

Microsoft (R) Macro Assembler Version 5.10
9/13/22 13:31:37

Symbols-1

Segments and Groups:

	N a m e	Length	Align
Combine Class			

```

        DGROUP . . . . . GROUP
        _DATA . . . . . 0047WORD
PUBLIC _DATA
        STACK . . . . . 0100 PARASTACK
        'STACK'
        TEXT . . . . . 0010WORD
PUBLIC _CODE

```

Symbols:

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

47464 + 459796 Bytes symbol space free

0 Warning Errors

Приложение D

Текст диагностического файла hello2

Microsoft (R) Macro Assembler Version 5.10
9/13/22 15:25:55

Page 1-1

```

1          ; HELLO2 - PJC+PμP±PSP°CЦ
PiCЪPsPiCЪP°P

          jPjP° N2 P»P°P±.CЪP°P±.#1 PïPs
PrPëCГC

          †PëPïP»PëPSPμ
"PђCЪC...PëC, PμPëC, CГCЪP° P

          ePsPjPiCЪCЪC, PμCЪP°"

2          ;          PμCЪPsPiCЪP°PjPjP°
PëCГPïPsP

          »CЪP·CГPμC, PiCЪPsC†PμPrCГCЪCГ

PrP»CЦ P

          ïPμC†P°C, Pë CГC, CЪPsPëPë

3          ;

4          ;          PŸP•PљPŸPŸ PμP PђP"P
PђPђPђP«

5

6 = 0024          EOFLine          EQU          '$'
; PђPiCЪPμPrP

```

```

                                µP»PµPSPëPµ
CÍPëPjPIPsP»CßPSPsPN° PePsPS

                                CÍC,P°PSC,C<

7                                ;
"PßPsPS

                                PµC† CÍC,CßPsPePë"

8

9                                ; PŸC,PµPe PìCßPsPìCßP°PjPjC<

10

11                                ASSUME CS:CODE, SS:AStack

12

13 0000                                AStack                SEGMENT
STACK

14 0000 000C[                                DW 12
DUP('!')                                ; PñC,PIPsP

                                rPëC,CÍCµ                12                CÍP»PsPI
PìP°PjCµC,Pë

15 0021

16                                ]

17

18 0018                                AStack                ENDS

19

20                                ;                                P"P°PSPSC< Pµ
PìCßPsPìCßP°PjPjC<

21

22 0000                                DATA                SEGMENT

23

```

```

24                                ;                P" PëCтPμPεC, PëPIC<
PsPïPëCíP°PSPëCи

                                PrP°PSPSC<C...

25

26 0000    48 65 6C 6C 6F 20    HELLO                DB
'Hello Worlds!', 0AH, 0DH,

                                EOFLine

27                57 6F 72 6C 64 73

28                21 0A 0D 24

29 0010    53 74 75 64 65 6E    GREETING                DB
'Student from 4350 - $'

30                74 20 66 72 6F 6D

31                20 34 33 35 30 20

32                2D 20 24

33 0025                                DATA                ENDS

34

35                                ; PђPsPr PïCтPsPïCтP°PjPjC<

36

37 0000                                CODE                SEGMENT

38                                ;                PұCтPsC†PμPrCíCтP°
PïPμC†P°C, Pë CíC, C

                                тPsPePë

39 0000                                WriteMsg    PROC    NEAR

40 0000    B4 09                                mov     AH, 9

41 0002    CD 21                                int     21h    ;
P'C<P·PsPI C,,CíP

```

	SPeC†PëPë	DOS	PiPs
PiCtPµCtC< PIP°PSPëCt			
Microsoft (R) Macro Assembler Version 5.10			
9/13/22 15:25:55			
Page 1-2			
42 0004 C3		ret	
43 0005	WriteMsg	ENDP	
44			
45	;	P`PsP»PsPIPSP°Ct	
PiCtPµC†PµPrC†CtP°			
46 0005	Main	PROC FAR	
47 0005 1E		push DS	
; \ PŸPsC...CtP°			
PSP°C†P°P»P° PS	PSPµPSPëPµ	P°PrCtPµC†P°	
	P PI C†C, PµPePµ		
48 0006 2B C0		sub AX, AX	
; > PrP»Ct PiP			
	sC†P»PµPrC†CtC%PµPiPs		
PIPsC†C†C, P°PSPsP			
	IP»PµPSPëCt PiPs		
49 0008 50		push AX	
; / PePsPjP°PS			
	PrPµ	ret,	
P·P°PIPµCtCëP°CtC%PµPN° PiCtPµ			
	C†PµPrC†CtC†.		
50 0009 B8 ---- R		mov	
AX, DATA	; P		

```

                                -P°PiCṪCíP·PeP°
CíPµPiPjPµPSC,PSPsPiPs
                                51 000C 8E D8                                mov    DS,AX
; C
                                ṪPµPiPḔCíC,CṪP° PrP°PSPSC<C...
                                52 000E BA 0000 R                                mov    DX,
OFFSET HELLO ; P
                                'C<PIPsPr PSP° ĆPeCṪP°PS
PiPµCṪPIPsPN°
                                53 0011 E8 0000 R                                call
WriteMsg ; C
                                ÍC,CṪPsPePḔ
PiCṪPḔPIPµC,CíC,PIPḔCṪ.
                                54 0014 BA 0010 R                                mov    DX,
OFFSET GREETING ; P
                                'C<PIPsPr PSP° ĆPeCṪP°PS
PIC,PsCṪPsPN°
                                55 0017 E8 0000 R                                call
WriteMsg ; C
                                ÍC,CṪPsPePḔ
PiCṪPḔPIPµC,CíC,PIPḔCṪ.
                                56 001A CB                                ret
; P
                                'C<C...PsPr PI DOS PiPs
PePsPjP°PSPPrPµ,
                                57
; P
                                SP°C...PsPrCṪC%PµPN°CíCṪ PI 1-PsPj
CíP»PsP
                                IPµ PSP.
                                58 001B                                Main ENDP

```

```

59 001B                                CODE      ENDS
60                                END Main
Microsoft (R) Macro Assembler Version 5.10
9/13/22 15:25:55

```

Symbols-1

Segments and Groups:

Combine Class	N a m e	Length	Align
	ASTACK	0018	PARASTACK
	CODE	001B	PARANONE
	DATA	0025	PARANONE

Symbols:

	N a m e	Type	Value	Attr
	EOFLINE	NUMBER		0024
DATA	GREETING	L BYTE		0010
DATA	HELLO	L BYTE		0000
CODE	MAIN	F PROC		0005

Length = 0016

WRITEMSG N PROC 0000
CODELength = 0005

@CPU TEXT 0101h

@FILENAME TEXT hello2

@VERSION TEXT 510

52 Source Lines

52 Total Lines

13 Symbols

47456 + 459801 Bytes symbol space free

0 Warning Errors

0 Severe Errors