

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 Михаил
Александрович

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

```
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
DOS
        END

```

Текст исходной программы 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

```


8
PsP±PμCÍPîPμC†PëPIP°Pμ

CÍC,CṽPsP

9
P·P°PeP°PSC†PëPIP°CṽC%

10
P·P°PrP

PSPsPjPμ

11
CṽPμPîPëCÍC,CṽPμ

P°PrCṽPμCÍP°

12

13
PëCÍPîPSP»CṽP·CÍPμC, C

CÍPsC...CṽP°P

14
CÍPsPrPμCṽP¶PëPjPsPμ.

15

16

17

CÍPμPîPjPμPSC,

18

; -

C, PIC<PIPsPr PSP° CÍPeCṽP°PS

ePë CÍPëPjPIPsP»PsPI,

;

PμPN°CÍC¶ P·PSP°PePsPj "\$";

; - C,CṽPμP±CÍPμC,

°PSPëC¶ PI CṽPμPîPëCÍC,CṽPμ ah

CṽP° C,,CÍPSPeC†PëPë=09h,

; P° PI

dx - CÍPjPμC%PμPSPëC¶

PIC<PIPsPrPëPjPsPN°

; CÍC,CṽPsPePë;

; -

ṽPμPîPëCÍC,Cṽ ax Pë PSPμ

SC¶PμC, PμPîPs

;

;

DOSSEG

; P—P°PrP°PSPëPμ

PsPI PîPsPr P"PhPŸ

.MODEL SMALL

```

;  PḥPsPrPμP»CḤ

PἱP°PjCḶC,PḔ-SMA

                                LL(PḥP°P»P°CḶ)

19                                .STACK 100h

;  PḥC,PiPμCḶC,PḔ PἱPsPr

PŸC,PμP

                                € 256 P±P°PN°C,

20                                .DATA

;  PḱP°C±P°P»Ps

CḶPμPἱPjPμPSC,P°

                                PrP°PSPSC<C...

21 0000                                Greeting LABEL BYTE

;  PŸPμPeCḶC,

PἱCḬPḔPiPμC,CḶC,P

                                IPḔCḶ

22 0000  D0 92 D0 B0 D1 81                                DB 'P'P°CḶ
PἱCḬPḔPiPμC,CḶC,PiCḶPμC,

                                CḶC,.PiCḬ.7303 - P~PiP°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
                                     ;  PĶP°C†P°P»Ps  CÍPµPiPjPµPSC,P°
PePsPr
                                     P°
35 0000  B8 ---- R                  mov  ax, @data
                                     ;  P-P°PiCĤCÍP·PeP°  PI  DS
P°PrCĤPµCÍP°
                                     PSP°C†P°P»P°
36 0003  8E D8                      mov  ds, ax
                                     ;  CÍPµPiPjPµPSC,P°  PrP°PSPSC<C...
37 0005  BA 0000 R                  mov  dx, OFFSET Greeting
                                     ;  P-P°PiCĤCÍP·PeP°  PI  dx
CÍPjPµC%PµPS
                                     PëCĬ
38
                                     ;  P°PrCĤPµCÍP°  C,PµPeCÍC,P°
PiCĤPëPIPµ
                                     C,CÍC,PIPëCĬ
39 0008                      DisplayGreeting:
40 0008  B4 09                      mov  ah, 9
                                     ;  #  C„CÍPSPeC†PëPë  P"PhPŸ
PiPµC†P°C,P
                                     ë CÍC,CĤPsPePë
41 000A  CD 21                      int  21h
                                     ;  PIC<PIPsPr  PSP°  CĶPeCĤP°PS
PiCĤPëPI
                                     PµC,CÍC,PIPëCĬ
42 000C  B4 4C                      mov  ah, 4ch

```

```

; # C„CřPSPeC†PëPë P"PhPŸ
P·P°PIPµCȤC

€PµPSPëCȤ PìCȤPsPìCȤP°PjPjC<
43 000E CD 21 int 21h

; P·P°PIPµCȤC€PµPSPëPµ
PìCȤPsPìCȤP°PjP

jC< Pë PIC<C...PsPr PI P"PhPŸ
44 END

```

```

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

```

Symbols-1

Segments and Groups:

Combine Class	N a m e	Length	Align
	DGROUP	GROUP	
'DATA'	_DATA	0047 WORD PUBLIC	
	STACK	0100 PARA STACK 'STACK'	
'CODE'	_TEXT	0010 WORD PUBLIC	

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

47464 + 459796 Bytes symbol space free

0 Warning Errors

0 Severe Errors

Текст диагностического файла hello2.lst

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  PiPs
PrPëCfC

                                †PëPiP»PëPSPμ
"PђCЪC...PëC,PμPeC,CfCЪP° P

                                ePsPjPiCЪCЪC,PμCЪP°"

2                                ;                      PμCЪPsPiCЪP°PjPjP°
PëCfPiPsP

                                »CЪP·CfPμC,          PiCЪPsC†PμPrCfCЪCf
PrP»CЦ P

                                iPμC†P°C,Pë CfC,CЪPsPePë

3                                ;

4                                ;                      PŷP•PљPŷPŷ      PμP  PђP"P
PђPђPђP«

5

6 = 0024                        EOFLine  EQU  '$'                      ;
PђPiCЪPμPrP

                                μP»PμPSPëPμ      CfPëPjPiPsP»CЪPSPSPN°
PePsPS

                                CfC,P°PSC,C<

7                                ;

"PљPSPS
```

```

PμC† CÍC,CBPsPePë"

8
9          ; PŸC,PμPe  PìCBPsPìCBP°PjPjC<
10
11          ASSUME CS:CODE, SS:AStack
12
13 0000          AStack      SEGMENT  STACK
14 0000  000C[          DW 12 DUP('!')      ;
PñC,PIPsP
          rPëC,CÍCÇ 12 CÍP»PsPI PìP°PjCÇC,Pë
15      0021
16          ]
17
18 0018          AStack      ENDS
19
20          ; P"P°PSPSC<Pμ PìCBPsPìCBP°PjPjC<
21
22 0000          DATA      SEGMENT
23
24          ;          P"PëCBPμPeC,PëPIC<
PsPìPëCÍP°PSPëCÇ
          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                                ; PŁPsPr PiCŁPsPiCŁP°PjPjC<
36
37 0000                                CODE      SEGMENT
38                                ; PŁCŁPsC†PŁPrC†CŁP° P†PŁC†P°C,PŁ
C†C,C                                ŁPsPePŁ

39 0000                                WriteMsg  PROC   NEAR
40 0000  B4 09                                mov    AH,9
41 0002  CD 21                                int     21h  ;
P' C< P·PsPI C„C†P                                SPeC†PŁPŁ DOS      P†Ps
PiCŁPŁCŁC< PIP°PSPŁCŁ                                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°CŁ
PiCŁPsC†PŁPrC†CŁP°                                Main      PROC   FAR
46 0005
47 0005  1E                                push    DS      ;\
PŸPsC...CŁP°                                PSPŁPSPŁPŁ P°PrCŁPŁC†P°
PSP°C†P°P»P° PS                                P PI C†C,PŁPePŁ
48 0006  2B C0                                sub     AX,AX    ; >
PrP»CŁ P†P

```

PIPsCfCfC, P°PSPsP	sCfP»PµPrCfCfC%PµPiPs
	IP»PµPSPëCf PïPs
49 0008 50	push AX ;/
PePsPjP°PS	
PiCfPs	PrPµ ret, P·P°PIPµCfC€P°CfC%PµPN°
	C†PµPrCfCfCf.
50 0009 B8 ---- R	mov AX, DATA
; P	
CfPµPiPjPµPSC, PSPsPiPs	-P°PiCfCfP·PeP°
51 000C 8E D8	mov DS, AX
; C	
	BPµPiPëCfC, CfP° PrP°PSPSC<C...
52 000E BA 0000 R	mov DX, OFFSET
HELLO ; P	
PiPµCfPIPsPN°	'C<PIPsPr PSP° CkPeCfP°PS
53 0011 E8 0000 R	call WriteMsg
; C	
	fC, CfPsPePë PiCfPëPIPµC, CfC, PIPëCf.
54 0014 BA 0010 R	mov DX, OFFSET
GREETING ; P	
PIC, PsCfPsPN°	'C<PIPsPr PSP° CkPeCfP°PS
55 0017 E8 0000 R	call WriteMsg
; C	
	fC, CfPsPePë PiCfPëPIPµC, CfC, PIPëCf.
56 001A CB	ret
; P	
PePsPjP°PSPPrPµ,	'C<C...PsPr PI DOS PïPs
57	
; P	
CfP»PsP	SP°C...PsPrCfC%PµPN°CfCf PI 1-PsPj

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	PARA STACK
	CODE	001B	PARA NONE
	DATA	0025	PARA NONE

Symbols:

	N a m e	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
```

```
52 Source Lines
52 Total Lines
13 Symbols
47456 + 459801 Bytes symbol space free
0 Warning Errors
0 Severe Errors
```

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 код команды	Содержимое регистров и ячеек памяти	
			до выполнения	после выполнения
0010	Mov AX, 1A07	B8071A	(AX) = 0000 (IP) = 0010 (Flags) = 7202 (SF) = 0	(AX) = 1A07 (IP) = 0013 (Flags) = 7280 (SF) = 1
0013	Mov DS, AX	8ED8	(AX) = 1A07 (DS) = 19F5 (IP) = 0013	(AX) = 1A07 (DS) = 1A07 (IP) = 0015
0015	Mov Ah, 09	B409	(IP) = 0015	(IP) = 0018
0018	INT 21	CD21	(AX) = 1A07 (IP) = 0018	(AX) = 0907 (IP) = 001A
001A	Mov Ah, 4C	B44C	(IP) = 001A	(IP) = 001C
001C	INT 21	CD21	(AX) = 0907 (IP) = 001C	(AX) = 4C07 (IP) = 001E
001E	Program terminated OK			

Табл.1. Протокол пошагового исполнения программы hello2

Адрес Команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			до выполнения	после выполнения
0005	Push DS	1E	(DS) = 19F5 (SP) = 0018 (IP) = 0005 (Stack) +0 0000 (Flags) = 7202	(DS) = 19F5 (SP) = 0016 (IP) = 0006 (Stack) +0 19F5 (Flags) = 7200
0006	Sub AX, AX	2BC0	(AX) = 0000 (IP) = 0006 (Flags) = 7200	(AX) = 0000 (IP) = 0008 (Flags) = 7244

			(ZF) = 0 (PF) = 0	(ZF) = 1 (PF) = 1
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 (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 (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 (Stack) +2 0000 (Stack) +4 19F5	(SP) = 0014 (IP) = 0014 (Stack) +0 0000 (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	(IP) = 0002
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 (Stack) +0 0000 (Stack) +2 19F5	(SP) = 0018 (IP) = 0000 (Stack) +0 0000 (Stack) +2 0000
0000	INT 20	CD20	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