

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

Студент гр. 1381

Сагидуллин Э.Р.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

Изучить основные принципы трансляции, научиться отлаживать и выполнять программы на языке Ассемблера.

Задание

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

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

Ход выполнения работы

В ходе выполнения лабораторной работы в каталог masm были загружены программы hello1.asm и hello2.asm. Далее программы были транслированы с созданием объектного файла и файла диагностических сообщений. Был скомпонован загрузочный файл с созданием исполняемого файла. Далее была выполнена программа, и произведена отладка в пошаговом режиме с помощью отладчика afd.

Таблица 1. Начальное значение регистров

CS	DS	ES	SS
1A05	19F5	19F5	1A0C

Таблица 2. Протокол работы программы hello1

Адрес команды	Символьный код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0010	MOV AX, 1A07	B8071A	(AX)=0000 (DS)=19F5 (IP)=0010	(AX)=1A07 (DS)=19F5 (IP)=0013
0013	MOV DS, AX	8ED8	(AX)=1A07 (DS)=19F5 (IP)=0013	(AX)=1A07 (DS)=1A07 (IP)=0015
0015	MOV DX, 0000	BA0000	(AX)=1A07 (DS)=1A07 (IP)=0015	(AX)=1A07 (DS)=1A07 (IP)=0018
0018	MOV AH, 09	B409	(AX)=1A07 (DS)=1A07 (IP)=0018	(AX)=0907 (DS)=1A07 (IP)=001A

001A	INT 21	CD21	(AX)=0907 (DS)=1A07 (IP)=001A	(AX)=0907 (DS)=1A07 (IP)=001C
001C	MOV AH, 4C	B44C	(AX)=0907 (DS)=1A07 (IP)=001C	(AX)=4C07 (DS)=1A07 (IP)=001E
001E	INT 21	CD21	(AX)=4C07 (DS)=1A07 (IP)=001E	(AX)=0000 (DS)=19F5 (IP)=0010

Таблица 3. Начальное значение регистров

CS	DS	ES	SS
1A05	19F5	19F5	1A0C

Таблица 4. Протокол работы программы hello2

Адрес команды	Символьный код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0005	PUSH DS	1E	(AX)=0000 (DS)=19F5 (IP)=0005 STACK +0 0000 +2 0000 +4 0000 +6 0000	(AX)=0000 (DS)=19F5 (IP)=0006 STACK +0 19F5 +2 0000 +4 0000 +6 0000
0006	SUB AX, AX	2BC0	(AX)=0000 (DS)=19F5 (IP)=0006	(AX)=0000 (DS)=19F5 (IP)=0008
0008	PUSH AX	50	(AX)=0000 (DS)=19F5 (IP)=0008 STACK +0 19F5 +2 0000 +4 0000 +6 0000	(AX)=0000 (DS)=19F5 (IP)=0009 STACK +0 0000 +2 19F5 +4 0000 +6 0000
0009	MOV AX, 1A07	B8071A	(AX)=0000 (DS)=19F5 (IP)=0009	(AX)=1A07 (DS)=19F5 (IP)=000C
000C	MOV DS, AX	8ED8	(AX)=1A07 (DS)=19F5 (IP)=000C	(AX)=1A07 (DS)=1A07 (IP)=000E
000E	MOV DX, 0000	BA0000	(AX)=1A07 (DS)=1A07 (IP)=000E	(AX)=1A07 (DS)=1A07 (IP)=0011
0011	CALL 0000	E8ECFF	(AX)=4C07 (DS)=1A07	(AX)=1A07 (DS)=1A07

			(IP)=001E STACK +0 0000 +2 19F5 +4 0000 +6 0000	(IP)=0000 STACK +0 0014 +2 0000 +4 19F5 +6 0000
0000	MOV AH, 09	B409	(AX)=1A07 (DS)=1A07 (IP)=0000	(AX)=0907 (DS)=1A07 (IP)=0002
0002	INT 21	CD21	(AX)=0907 (DS)=1A07 (IP)=0002	(AX)=0907 (DS)=1A07 (IP)=0004

0004	RET	C3	(AX)=0907 (DS)=1A07 (IP)=0004 STACK +0 0014 +2 0000 +4 19F5 +6 0000	(AX)=0907 (DS)=1A07 (IP)=0014 STACK +0 0000 +2 19F5 +4 0000 +6 0000
0014	BA1000	MOV DX, 0010	(AX)=0907 (DS)=1A07 (IP)=0014 (DX)=0000	(AX)=0907 (DS)=1A07 (IP)=0017 (DX)=0010
0017	E8E6FF	CALL 0000	(AX)=0907 (DS)=1A07 (IP)=0017 STACK +0 0000 +2 19F5 +4 0000 +6 0000	(AX)=0907 (DS)=1A07 (IP)=0000 STACK +0 001A +2 0000 +4 19F5 +6 0000
0000	MOV AH, 09	B409	(AX)=0907 (DS)=1A07 (IP)=0000	(AX)=0907 (DS)=1A07 (IP)=0002
0002	INT 21	CD21	(AX)=0907 (DS)=1A07 (IP)=0002	(AX)=0907 (DS)=1A07 (IP)=0004
0004	RET	C3	(AX)=0907 (DS)=1A07 (IP)=0004 STACK +0 001A +2 0000 +4 19F5 +6 0000	(AX)=0907 (DS)=1A07 (IP)=001A STACK +0 0000 +2 19F5 +4 0000 +6 0000
001A	RET Far	CB	(AX)=0907 (DS)=1A07 (IP)=001A STACK +0 0000 +2 19F5 +4 0000 +6 0000 (CS) = 1A0A	(AX)=0907 (DS)=1A07 (IP)=0000 STACK +0 0000 +2 0000 +4 0000 +6 0000 (CS) = 19F5
0000	INT 20	CD20	(AX)=0907 (DS)=1A07 (IP)=0000 (CS)=19F5	(AX)=00000 (DS)=19F5 (IP)=0005 (CS)=1A0A

Вывод

При выполнении лабораторной работы были изучены трансляция, отладка и выполнение программы на языке Assembler.

ПРИЛОЖЕНИЕ А

ТЕКСТЫ ИСХОДНЫХ ФАЙЛОВ

Hello1.asm

:

```
DOSSEG
.MODEL    SMALL
.STACK    100h
.DATA
Greeting LABEL BYTE                ; Текст
приветствия
    DB 'Вас приветствует ст.гр.1381 - Сагидуллин Э.Р.',13,10,'$'
.CODE
    mov ax, @data
    mov ds, ax
    mov dx, OFFSET Greeting

DisplayGreeting:
    mov ah, 9h
    int 21h
    mov ah, 4ch
    int 21h
END
```

Hello2.asm:

```
EOFLine EQU '$'
ASSUME CS:CODE, SS:AStack

AStack    SEGMENT STACK
           DW 12 DUP('!')
AStack    ENDS

DATA      SEGMENT

HELLO     DB 'Hello Worlds!', 0AH, 0DH, EOFLine
GREETING  DB 'Student from 1381 - $'
DATA      ENDS

CODE      SEGMENT
WriteMsg  PROC NEAR
           mov AH, 9
           int 21h
           ret
WriteMsg  ENDP

Main      PROC FAR
           push DS
```


sub AX,AX

```

        push    AX
        mov     AX, DATA
        mov     DS, AX
        mov     DX, OFFSET HELLO
        call    WriteMsg
        mov     DX, OFFSET GREETING
        call    WriteMsg
        ret

Main     ENDP
CODE     ENDS
        END Main

```

ПРИЛОЖЕНИЕ Б

ФАЙЛЫ ДИАГНОСТИЧЕСКИХ СООБЩЕНИЙ

Hello1.lst:

```

Microsoft      (R)      Macro      Assembler      Version      5.10
9/11/22  20:07:10

age      1-1

;      HELLO2      -      PJC+PμP±PSP°Cμ
PìCЪPsPìCЪP°PjPjP° N2
P»P°P±.CЪP°P±.#1      PìPs
PгPëCГC†PëPìP»PëPSPμ
hъCЪC...PëC, PμPеC, CГCЪP°
"P
;      PμCЪPsPìCЪP°PjPjP°
PеPsPjPìCЪCЪC, PμCЪP°"
μC, PìCЪPsC†PμPгCГCЪCГ PгP»Cμ PìPμC†P°C, Pë
PëCГPìPsP»CЪP·CГP
CЪPsPеPë
CГC,
;
;      PŷP•PъPŷPŷ PμP PђP"P PђPђPђP«
= 0024      EOFLine      EQU      '$'      ;
PђPìCЪPμPгPμP»PμPSP
ëPμ      CГPëPjPìPsP»CЪPSPsPN
PеPsPSCГC, P°PSC, C<
;
"PъPsPSPμC† CГC
,CЪPsPеPë"
; PŷC, PμPе PìCЪPsPìCЪP°PjPjC<
ASSUME CS:CODE, SS:AStack
0000      AStack      SEGMENT      STACK
0000  000C[      DW  12  DUP('!')      ;
PђC, PìPsPгPëC, CГC

```



```

                                0021      12 CÍP»PsPI PíP°PjC¼C,Pë
                                ]

0018      AStack      ENDS

                                ; P°P°PSPSC<Pµ PíC¼PsPíC¼P°PjPjC<

0000      DATA      SEGMENT

                                ;      P°PëC¼PµPeC,PëPIC< PsPíPëCÍP°PSPëC¼
PrP°PSPS
                                C<C...

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 1381 - $'
                                74 20 66 72 6F 6D
                                20 34 33 35 30 20
                                2D 20 24
0025      DATA      ENDS

                                ; PµPsPr PíC¼PsPíC¼P°PjPjC<

0000      CODE      SEGMENT
                                ;      PµC¼PsC†PµPrCíC¼P°      PíPµC†P°C,Pë
CÍC,C¼PsPePë
                                0000      WriteMsg  PROC  NEAR
                                0000      B4 09      mov    AH,9
                                0002      CD 21      int    21h    ; P'C<P·PsPI
C,,CíPSPeC†PëP
                                ë DOS PíPs PíC¼PµC¼C<PIP°PSPëC¼
                                0004      C3      ret
                                0005      WriteMsg  ENDP

                                ; P°PsP»PsPIPSP°C¼ PíC¼PsC†PµPrCíC¼P°

```

P

age 1-2

```

CtP°PSPµPSPë
                                Pµ P°PrCtPµCÍP° PSP°C†P°P»P° PSP PI
CÍC, PµPePµ
                                0006 2B C0 sub AX,AX ; > PrP»Cµ
PìPsCÍP»PµP
                                rCÍCtC%PµPìPs PìPsCÍCÍC, P°PSPsPIP»PµPSPëCµ
PìPs
                                0008 50 push AX ;/
PePsPjP°PSPPrPµ ret
                                , P·P°PIPµCtCëP°CtC%PµPN°
PìCtPµsC†PµPrCÍCtCÍ.
                                0009 B8 ---- R mov AX, DATA ;
P—P°PìCtC
                                íP·PeP° CÍPµPìPjPµPSC, PSPsPìPs
                                000C 8E D8 mov DS, AX ;
CtPµPìPëC
                                ÍC, CtP° PrP°PSPSC<C...
                                000E BA 0000 R mov DX, OFFSET HELLO ;
P'C<PIPsP
                                r PSP° CÍPeCtP°PS PìPµCtPìPsPN°
                                0011 E8 0000 R call WriteMsg ;
CÍC, CtPµsP
                                ëPë PìCtPëPIPµC, CÍC, PìPëCµ.
                                0014 BA 0010 R mov DX, OFFSET GREETING ;
P'C<PIPsP
                                r PSP° CÍPeCtP°PS PIC, PsCtPµsPN°
                                0017 E8 0000 R call WriteMsg ;
CÍC, CtPµsP
                                ëPë PìCtPëPIPµC, CÍC, PìPëCµ.
                                001A CB ret ;
P'C<C...PsP
                                r PI DOS PìPs PePsPjP°PSPPrPµ,

```

```

;
PSP°C...PsP
                                rC¼C%PµPN°CÍC¼ PI 1-PsPj CÍP»PsPIPµ PSP.
001B                                Main            ENDP
001B                                CODE            ENDS
                                END Main
Microsoft      (R)      Macro      Assembler      Version      5.10
9/11/22 20:07:10
S
ymbols-1
```

Segments and Groups:

Class	N a m e	Length	Align	Combine
	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
= 0016	MAIN	F PROC	0005	CODE Length
= 0005	WRITEMSG	N PROC	0000	CODE Length
	@CPU	TEXT	0101h	
	@FILENAME	TEXT	hello2	
	@VERSION	TEXT	510	

52 Source Lines
52 Total Lines
13 Symbols

47986 + 459271 Bytes symbol space free

0 Warning Errors
0 Severe Errors

Hello2.lst:

Microsoft (R) Macro Assembler Version 5.10
9/11/22 20:07:10

age 1-1

```
; HELLO2 - PJC+PμP±PSP°Cμ
PìCḂPsPìCḂP°PjPjP° N2
                                P»P°P±.CḂP°P±.#1 PìPs
PrPëCíC†PëPìP»PëPSPμ
ḡCḂC...PëC,PμPeC,CíCḂP°
"P
                                PμCḂPsPìCḂP°PjPjP°
PePsPjPìCḂCḂC,PμCḂP°"
μC, PìCḂPsC†PμPrCíCḂCí PrP»Cμ PìPμC+P°C,Pë
PëCíPìPsP»CḂP·CíP
CḂPsPePë
CíC,
;
; PŸP•PḂPŸPŸ PμP PḡP`P PḡPḂPḂP«
= 0024 EOFLine EQU '$' ;
PḡPìCḂPμPrPμP»PμPSP
ëPμ CíPëPjPìPsP»CḂPSPsPN°
PePsPSCíC,P°PSC,C<
;
"PḂPsPSPμC† CíC
,CḂPsPePë"
```



```

; PŸC,PuPe PiCᄁPsPiCᄁP°PjPjC<

ASSUME CS:CODE, SS:AStack

0000          AStack      SEGMENT  STACK
0000  000C[          DW  12  DUP('!')          ;
PhC,PIPsPrPëC,CÍC

          12 CÍP»PsPI PiP°PjCᄁC,Pë
0021

]

0018          AStack      ENDS

; P°P°PSPSC<Pu PiCᄁPsPiCᄁP°PjPjC<

0000          DATA      SEGMENT

;      P°PëCᄁPuPeC,PëPIC< PsPiPëCÍP°PSPëCᄁ
PrP°PSPS

C<C...

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 1381 - $'
          74 20 66 72 6F 6D
          20 34 33 35 30 20
          2D 20 24
0025          DATA      ENDS

; PᄁPsPr PiCᄁPsPiCᄁP°PjPjC<

0000          CODE      SEGMENT
;      PuCᄁPsC†PuPrCÍCᄁP°      PiPuC†P°C,Pë
CÍC,CᄁPsPePë
0000          WriteMsg PROC NEAR
0000  B4 09          mov    AH,9
0002  CD 21          int    21h      ; P'C<P·PsPI
C,,CÍPSPeC†PëP

ë DOS PiPs PiCᄁPuCᄁC<PIP°PSPëCᄁ

```



```

0004 C3 ret
0005 WriteMsg ENDP

; P"PsP»PsPIPSP°C¼ PïCßPsC†PµPrC†CßP°
Microsoft (R) Macro Assembler Version 5.10
9/11/22 20:07:10

age 1-2

CßP°PSPµPSPë
Pµ P°PrCßPµC†P° PSP°C†P°P»P° PSP PI
C†C, PµPePµ
0006 2B C0 sub AX, AX ; > PrP»C¼
PïPsC†P»PµP
rC†CßC%PµPïPs PIPsC†C†C, P°PSPsPIP»PµPSPëC¼
PïPs
0008 50 push AX ;/
PePsPjP°PSPPrPµ ret
, P·P°PIPµCßCëP°CßC%PµPN°
PïCßPsC†PµPrC†CßC†.
0009 B8 ---- R mov AX, DATA ;
P-P°PïCßC
íP·PeP° C†PµPïPjPµPSC, PSPsPïPs
000C 8E D8 mov DS, AX ;
CßPµPïPëC
íC, CßP° PrP°PSPSC<C...
000E BA 0000 R mov DX, OFFSET HELLO ;
P'C<PIPsP
r PSP° C†PeCßP°PS PïPµCßPIPsPN°
0011 E8 0000 R call WriteMsg ;
C†C, CßPsP
ëPë PïCßPëPIPµC, C†C, PIPëC¼.
0014 BA 0010 R mov DX, OFFSET GREETING ;
P'C<PIPsP
r PSP° C†PeCßP°PS PIC, PsCßPsPN°
0017 E8 0000 R call WriteMsg ;
C†C, CßPsP

```

еРё РїСѢРёРІРµС, СЃС, РІРёСЦ.

```

001A CB ret ;
P'C<C...PsP
r PI DOS PiPs PePsPjP°PSPrPu,
;
PSP°C...PsP
rC¼C%PuPN°CÍC¼ PI 1-PsPj CÍP»PsPIPu PSP.
001B Main ENDP
001B CODE ENDS
END Main
Microsoft (R) Macro Assembler Version 5.10
9/11/22 20:07:10
S
symbols-1

```

Segments and Groups:

Class	N a m e	Length	Align	Combine
	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
```

```
47986 + 459271 Bytes symbol space free
```

```
0 Warning Errors
0 Severe Errors
```