

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

Васильева О. М.

Преподаватель

Ефремов М. А.

Санкт-Петербург

2022

Задание.

Часть 1

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

Протокол работы на компьютере.

1. Трансляция программы с помощью строки

> masn lr1.asm

для создания объектного файла, файла диагностических сообщений (файла листинга) и файла перекрестных ссылок.

2. Компоновка загрузочного модуля с помощью строки

> link имя_файла.obj

3. Выполнение программы:

- в автоматическом режиме путем набора командной строки

> имя_файла.exe

- в пошаговом режиме под управлением отладчика

> afd имя_файла.exe

+ с фиксацией содержимого используемых регистров и ячеек памяти до и после выполнения команды.

Протокол пошагового исполнения.

Табл. 1- Hello1.exe

(DS) = 19F5; (SS) = 1A0C; (ES) = 19F5; (CS) = 1A05

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0010	B8071A	MOV AX, 1A07	(AX) = 0000 (IP) = 0010 (DS) = 19F5	(AX) = 1A07 (IP) = 0013 (DS) = 19F5
0013	8ED8	MOV DS, AX	(AX) = 1A07 (IP) = 0013 (DS) = 19F5	(AX) = 1A07 (IP) = 0015 (DS) = 1A07
0015	BA0000	MOV DX, 0000	(AX) = 1A07 (IP) = 0015 (DS) = 1A07	(AX) = 1A07 (IP) = 0018 (DS) = 1A07
0018	B409	MOV AH, 09	(AX) = 1A07 (IP) = 0018 (DS) = 1A07	(AX) = 0907 (IP) = 001A (DS) = 1A07
001A	CD21	INT 21	(AX) = 0907 (IP) = 001A (DS) = 1A07	(AX) = 0907 (IP) = 001C (DS) = 1A07
001C	B44C	MOV AH, 4C	(AX) = 0907 (IP) = 001C	(AX) = 4C07 (IP) = 001E

			(DS) = 1A07	(DS) = 1A07
001E	CD21	INT 21	(AX) = 4C07 (IP) = 001E (DS) = 1A07	(AX) = 0000 (IP) = 0010 (DS) = 19F5

Табл. 2 – Hello2.exe

(DS) = 19F5; (SS) = 1A05; (ES) = 19F5; (CS) = 1A0A

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

000C	MOV DS, AX	8ED8	(AX) = 1A07 (IP) = 000C (DS) = 19F5	(AX) = 1A07 (IP) = 000E (DS) = 1A07
000E	MOV DX, 0000	BA0000	(AX) = 1A07 (IP) = 000E (DS) = 1A07	(AX) = 1A07 (IP) = 0011 (DS) = 1A07
0011	CALL 0000	E8ECFF	(AX) = 1A07 (IP) = 0011 (SP) = 0014 STACK (+2) = 19F5 STACK (+4) = 0000 STACK (+0) = 0000	(AX) = 1A07 (IP) = 0000 (SP) = 0012 STACK (+2) = 0000 STACK (+4) = 19F5 STACK (+0) = 0014
0000	MOV AH, 09	B409	(AX) = 1A07 (IP) = 0000 (DS) = 1A07	(AX) = 0907 (IP) = 0002 (DS) = 1A07
0002	INT 21	CD21	(AX) = 0907 (IP) = 0002 (DS) = 1A07	(AX) = 0907 (IP) = 0004 (DS) = 1A07
0004	RET	C3	(AX) = 0907 (IP) = 0004 (SP) = 0012 STACK (+0) = 0014 STACK (+2) = 0000 STACK (+4) = 19F5	(AX) = 0907 (IP) = 0014 (SP) = 0014 STACK(+0) = 0000 STACK (+2) = 19F5 STACK (+4) = 0000
0014	MOV DX, 0010	BA1000	(AX) = 0907 (IP) = 0014 (DX) = 0000	(AX) = 0907 (IP) = 0017 (DX) = 0010

0017	CALL 0000	E8E6FF	(AX) = 0907 (IP) = 0017 (SP) = 0014 STACK (+0) = 0000 STACK (+2) = 19F5 STACK (+4) = 0000	(AX) = 0907 (IP) = 0000 (SP) = 0012 STACK (+0) = 001A STACK (+2) = 0000 STACK (+4) = 19F5
0000	MOV AX, 09	B409	(AX) = 0907 (IP) = 0000 (DS) = 1A07	(AX) = 0907 (IP) = 0002 (DS) = 1A07
0002	INT 21	CD21	(AX) = 0907 (IP) = 0002 (DS) = 1A07	(AX) = 0907 (IP) = 0004 (DS) = 1A07
0004	RET	C3	(AX) = 0907 (IP) = 0004 (DS) = 1A07 STACK (+0) = 001A STACK (+2) = 0000 STACK (+4) = 19F5	(AX) = 0907 (IP) = 001A (DS) = 1A07 STACK (+0) = 0000 STACK (+2) = 19F5 STACK (+4) = 0000
001A	RET FAR	CB	(AX) = 0907 (IP) = 001A (SP) = 0012 (CS) = 1A0A STACK (+2) = 19F5	(AX) = 0907 (IP) = 0000 (SP) = 0018 (CS) = 19F5 STACK (+2) = 0000

0000	INT 20	CD20	(AX) = 0907 (IP) = 0000 (DS) = 1A07 (DX) = 0010	(AX) = 0000 (IP) = 0005 (DS) = 19F5 (DX) = 0000
------	--------	------	--	--

Вывод.

Мы освоили работу с ассемблером `masm.exe`, линкером `link.exe` и отладчиком `afd.exe`. Скомпилировали программу, которая выводит сообщение на экран, и разобрались в ее структуре.

ПРИЛОЖЕНИЕ А. Код программ.

Имя файла: hello1.asm

```
DOSSEG ; Задание сегментов под
DOS
.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.1381 - Васильева О.М.',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 DOS печати строки
int 21h ; вывод на экран приветствия
mov ah, 4ch ; # функции DOS завершения про-
граммы
int 21h ; завершение программы и выход в
DOS
END
```

Имя файла: hello2.asm

```
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 В. Листинг успешной трансляции программы.

Текст сообщения (hello1.lst).

Microsoft (R) Macro Assembler Version 5.10
17:35:15

9/14/22

Page 1-1

```
; HELLO1.ASM - CГPиCтPсC%PμPSPSP°Cμ PIPμCтCГPё
Cμ CГC†PμP±PSPSP№ PиCтPсPиCтP°PjPjC< P»P°P±.CтP
°P±. N1
;
; PиPs PГPёCГC†PёPиP»PёPSPμ "PђCт
C...PёC, PμPеC, CГCтP° PеPсPjPиCтCтC, PμCтP°"
; *****
; *****
; PКP°P·PSP°C†PμPSPёPμ: PμCтPсPиCтP°PjPjP° C„Ps
CтPjPёCтCГPμC, Pё PIC<PIPsPГPёC, PSP° CКPеCтP°P
S PиCтPёPIPμC, CГC, PIPёPμ
;
; PиPsP»CтP·PsPIP°C, PμP»Cμ CГ PиPsP
jPсC%тCт C„CГPSPеC†PёPё P"PhPŸ "P'C<PIPsPГ CГC
, CтPсPеPё"
;
; (PSPсPjPμCт 09 PиCтPμCтC<PIP°PSPё
Pμ 21h), PеPсC, PsCтP°Cμ:
;
; - PsP±PμCГPиPμC†PёPIP°PμC, PIC<P
IPsPГ PSP° CКPеCтP°PS CГC, CтPсPеPё CГPёPjPIPsP»
PsPI,
;
; P·P°PеP°PSC†PёPIP°CтC%PμP№CГCμ
P·PSP°PеPsPj "$";
;
; - C, CтPμP±CГPμC, P·P°PГP°PSPёCμ
PI CтPμPиPёCГC, CтPμ ah PSPсPjPμCтP° C„CГPSPеC†P
ёPё=09h,
;
; P° PI CтPμPиPёCГC, CтPμ dx - C
ГPjPμC%PμPSPёCμ P°PГCтPμCГP° PIC<PIPsPГPёPjPs
P№
;
; CГC, CтPсPеPё;
;
; - PёCГPиPsP»CтP·CГPμC, CтPμPиPёC
ГC, Cт ax Pё PSPμ CГPсC...CтP°PSCμPμC, PμPiPs;
; CГPsPГPμCтP¶PёPjPsPμ.
; *****
; *****
```

```
DOSSEG
; P—P°PГP°PSPёPμ CГPμPиPjPμPSC, PsPI PиPsPГ P"Ph
PŸ
```

```
.MODEL SMALL
; PтPsPГPμP»Cт PиP°PjCμC, Pё-SMALL(PтP°P»P°Cμ)
.STACK 100h
; PhC, PIPμCГC, Pё PиPsPГ PŸC, PμPе 256 P±P°P№C,
.DATA
; PКP°C†P°P»Ps CГPμPиPjPμPSC, P° PГP°PSPSC<C...
Greeting LABEL BYTE
; PŸPμPеCГC, PиCтPёPIPμC, CГC, PIPёCμ
```

0000 DB 'P'P°CГ PиCтPёPIPμC, CГC, PICГPμC,
CГC, .PiC

```
т.7303 - PμPIP°PSPsPI Pμ.Pμ.',13,10,'$'
```

20 D0 BF D1 80 D0

```

B8 D0 B2 D0 B5 D1
82 D1 81 D1 82 D0
B2 D1 83 D0 B5 D1
82 20 D1 81 D1 82
2E D0 B3 D1 80 2E
37 33 30 33 20 2D

```

Microsoft (R) Macro Assembler Version 5.10
17:35:15

9/14/22

Page 1-2

```

20 D0 98 D0 B2 D0
B0 D0 BD D0 BE D0
B2 20 D0 98 2E D0
98 2E 0D 0A 24

```

```

                                .CODE                                ; P°P°C
                                ‡P°P»Ps C°PµP°PjPµPSC,P° P°P°SP°P°
0000 B8 ---- R                mov ax, @data                        ; P-P°
                                PiC°C°P·P°P° PI DS P°P°C°PµC°P° PSP°C°P°P»P°
0003 8E D8                    mov ds, ax                            ;
C°Pµ
                                PiPjPµPSC,P° P°P°PSPSC<C...
0005 BA 0000 R                mov dx, OFFSET Greeting              ; P-P°
                                PiC°C°P·P°P° PI dx C°PjPµC°PµPSP°Cµ
                                ; P°P°C
                                PµPµC°P° C,PµP°C°P° P°P°P°P°PµC,C°P°P°P°Cµ
0008                                DisplayGreeting:
0008 B4 09                    mov ah, 9                            ; #
C,,
                                C°PSP°C°P°P°P° P°P°P°P° P°PµC°P°C,P° C°P°C,C°P°P°P°P°
000A CD 21                    int 21h                                ;
PIC<P
                                IPS°P° PSP° C°P°C°P°PS P°P°P°P°PµC,C°P°P°P°Cµ
000C B4 4C                    mov ah, 4ch                            ; #
C,,
                                C°PSP°C°P°P°P° P°P°P°P° P·P°P°PµC°C°PµPSP°Cµ P°P°P°S
                                PiC°P°PjPjC<
000E CD 21                    int 21h                                ;
P·P°P
                                IPµC°C°PµPSP°Pµ P°P°P°P°P°P°PjPjC< P° PIC<C...P°P° P° PI P°P°P°P°
                                END

```

Microsoft (R) Macro Assembler Version 5.10
17:35:15

9/14/22

Symbols-1

Segments and Groups:

	N a m e	Length	Align	Combine	Class
DGROUP		GROUP		
_DATA	0047	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 + 459266 Bytes symbol space free

```

0 Warning Errors
0 Severe Errors

```

Текст сообщения (hello2.lst).

Microsoft (R) Macro Assembler Version 5.10
9/11/22 11:05:43

Page

1-1

```

; HELLO2 - PJC†PμP±PSP°CЎ PİCЊPsPİCЊP°PjPjP°
N2      P»P°P±.CЊP°P±.#1 PİPs PŀPЄCŀC†PЄPİP»PЄPSPμ
"p      ħCЊC...PЄC,PμPєC,CŀCЊP° PєPSPjPİCЊCЊC,PμCЊP°"
;      PЎCЊPsPİCЊP°PjPjP°
PЄCŀPİPSP»CЊP·CŀP

```

```

CfC,                                µC, PīCtPsc†PµPŕCŕCtCŕ PŕP»Cµ PīPµC†P°C,Pë
                                     CtPscPePë
                                     ;
                                     ; PŷP•PbPŷPŷ PµP PħP"P PĥPbPbP«

= 0024                                EOFLine EQU '$' ;
PħPīCtPµPŕPµP»PµPSP                ëPµ CŕPëPjPIPsP»CtPSPsPµ PePsPSCŕC,P°PSC,C<
                                     ; "PbPSPSPµC†

CfC                                    ,CtPscPePë"

                                     ; PŷC,PµPe PīCtPscPiCtP°PjPjC<

ASSUME CS:CODE, SS:AStack

0000                                AStack SEGMENT STACK
0000 000C[                            DW 12 DUP('!') ;
PħC,PiPsPŕPëC,CfC
0021                                µ 12 CŕP»PsPI PīP°PjCµC,Pë
]

0018                                AStack ENDS

                                     ; P"P°PSPSC<Pµ PīCtPscPiCtP°PjPjC<

0000                                DATA SEGMENT

                                     ; P"PëCtPµPeC,PëPIC< PsPīPëCŕP°PSPëCµ
PŕP°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 4350 - $'
74 20 66 72 6F 6D
20 34 33 35 30 20
2D 20 24
0025                                DATA ENDS

                                     ; PbPSPŕ PīCtPscPiCtP°PjPjC<

0000                                CODE SEGMENT
                                     ; PµCtPsc†PµPŕCŕCtP° PīPµC†P°C,Pë CfC,CtPscPePë
0000 WriteMsg PROC NEAR
0000 B4 09 mov AH,9
0002 CD 21 int 21h ; P'C<P•PsPI
C,,CŕPSPëC†PëP

ë DOS PīPs PīCtPµCtC<PIP°PSPëCt
0004 C3 ret
0005 WriteMsg ENDP

                                     ; P"PSP»PsPIPsP°Cµ PīCtPsc†PµPŕCŕCtP°

```

1-2

```

0005          Main      PROC   FAR
0005  1E          push   DS          ;\
PŸPsc...CŦP°PSPµPSPë

0006  2B C0      Pµ P°PŦCŦPµCŦP° PSP°CŦP°P»P° PSP PI CŦC,PµPëPµ
PİPscŦP»PµP      sub    AX,AX      ; > PŦP»CŦ

PİPs          ŦCŦCŦC%PµPİPs PİPscŦCŦC,P°PSPsPİP»PµPSPëCŦ

0008  50          push   AX          ;/ PëPspjP°PSPŦPµ
ret          , P·P°PİPµCŦCëP°CŦC%PµPŦ PİCŦPscŦPµPŦCŦCŦCŦ.

0009  B8 ---- R      mov    AX,DATA          ; P-
P°PİCŦC

000C  8E D8      ŦP·PëP° CŦPµPİPjPµPSC,PSPsPİPs
CŦPµPİPëC      mov    DS,AX          ;

000E  BA 0000 R      ŦC,CŦP° PŦP°PSPSC<C...
P'C<PİPSP      mov    DX, OFFSET HELLO      ;

0011  E8 0000 R      Ŧ PSP° CŦPëCŦP°PS PİPµCŦPİPsPŦ
CŦC,CŦPSP      call   WriteMsg          ;

0014  BA 0010 R      ëPë PİCŦPëPİPµC,CŦC,PİPëCŦ.
P'C<PİPSP      mov    DX, OFFSET GREETING ;

0017  E8 0000 R      Ŧ PSP° CŦPëCŦP°PS PİC,PscŦPSPŦ
CŦC,CŦPSP      call   WriteMsg          ;

001A  CB      ëPë PİCŦPëPİPµC,CŦC,PİPëCŦ.
P'C<C...PSP      ret          ;

PSP°C...PSP      Ŧ PI DOS PİPs PëPspjP°PSPŦPµ,
;

001B          ŦCŦC%PµPŦCŦCŦCŦ PI 1-Pspj CŦP»PspİPµ PSP.
001B          Main      ENDP
CODE          ENDS
END Main

```

bols-1

Segments and Groups:

	N a m e	Length	Align	Combine Class
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