

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

> masm 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	(AX) = 0907

			(IP) = 0017 (SP) = 0014 STACK (+0) = 0000 STACK (+2) = 19F5 STACK (+4) = 0000	(IP) = 0000 (SP) = 0012 STACK (+0) = 001A STACK (+2) = 0000 STACK (+4) = 19F5
0000	MOV AH, 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
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.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 ДОС печати строки
int 21h ; вывод на экран приветствия
mov ah, 4ch ; # функции ДОС завершения про-
граммы
int 21h ; завершение программы и выход в
ДОС
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 - CíPíCtPSc%PµPSPSP°Cµ PIPµCtCíPë
Cµ CíCtPµP±PSPSPNµ PíCtPSPíCtP°PjPjC< P»P°P±.CtP
°P±. N1
;
; PíPs PíPëCíCtPëPíP»PëPSPµ "PñCt
C...PëC, PµPëC, CíCtP° PëPSPjPíCtCtC, PµCtP°"
; *****
; *****
; PíP°P·PSP°CtPµPSPëPµ: PµCtPSPíCtP°PjPjP° C„Ps
CtPjPëCtCíPµC, Pë PíC<PíPSPíPëC, PSP° CíPëCtP°P
S PíCtPëPíPµC, CíC, PíPëPµ
;
; PíPSP»CtP·PSPíP°C, PµP»Cµ Cí PíPSP
jPSc%CtCt C„CíPSPëCtPëPë P”PñPŸ "P’C<PíPSPí CíC
,CtPSPëPë"
;
; (PSPSPjPµCt 09 PíCtPµCtC<PíP°PSPë
Pµ 21h), PëPSc, PScCtP°Cµ:
;
; - PSp±PµCíPíPµCtPëPíP°PµC, PíC<P
íPSPí PSP° CíPëCtP°PS CíC, CtPSPëPë CíPëPjPíPSP»
PSPI,
;
; P·P°PëP°PSCtPëPíP°CtC%PµPNµCíCµ
P·PSP°PëPSPj "$";
;
; - C, CtPµP±CíPµC, P·P°PíP°PSPëCµ
Pí CtPµPíPëCíC, CtPµ ah PSPSPjPµCtP° C„CíPSPëCtP
ëPë=09h,
;
; P° Pí CtPµPíPëCíC, CtPµ dx - C
íPjPµC%PµPSPëCµ P°PíCtPµCíP° PíC<PíPSPíPëPjPSP
PNµ
;
; CíC, CtPSPëPë;
;
; - PëCíPíPSP»CtP·CíPµC, CtPµPíPëC
íC, Ct ax Pë PSPµ CíPSc...CtP°PSCµPµC, PµPíPs;
; CíPSPíPµCtPñPëPjPSPµ.
; *****
; *****
```

```
DOSSEG
; P—P°PíP°PSPëPµ CíPµPíPjPµPSC, PSPI PíPSPí P”Pñ
PŸ
```

```
.MODEL SMALL
; PñPSPíPµP»Ct PíP°PjCµC, Pë-SMALL(PñP°P»P°Cµ)
.STACK 100h
; PñC, PíPµCíC, Pë PíPSPí PŸC, PµPë 256 P±P°PNµC,
.DATA
; PíP°CtP°P»Ps CíPµPíPjPµPSC, P° PíP°PSPSC<C...
Greeting LABEL BYTE
; PŸPµPëCíC, PíCtPëPíPµC, CíC, PíPëCµ
```

```
0000 DB 'P’P°Cí PíCtPëPíPµC, CíC, PíCíPµC,
CíC, .PíC
```

```
т.7303 - PµPíP°PSPSPí 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°PµPiPjPµPSC,P° PePsPr°P°
0000 B8 ---- R                mov ax, @data                        ; P-P°
                                PiC°C°P·PeP° PI DS P°Pr°C°PµC°P° PSP°C°P°P»P°
0003 8E D8                    mov ds, ax                            ;
C°Pµ
                                PiPjPµPSC,P° Pr°P°PSPSC<C...
0005 BA 0000 R                mov dx, OFFSET Greeting              ; P-P°
                                PiC°C°P·PeP° PI dx C°PjPµC°PµPSP°Cµ
                                ; P°Pr°C
                                °PµC°P° C,PµPeC°C,P° PiC°P°PIPµC,C°C,PiP°Cµ
0008 DisplayGreeting:
0008 B4 09                    mov ah, 9                            ; #
C,,
                                C°PSP°C°P°P°P° P°P°P°P° PiPµC°P°C,P° C°C,C°P°P°P°P°
000A CD 21                    int 21h                                ;
PIC<P
                                IP°Pr° PSP° C°K°P°C°P°PS PiC°P°PIPµC,C°C,PiP°Cµ
000C B4 4C                    mov ah, 4ch                            ; #
C,,
                                C°PSP°C°P°P°P° P°P°P°P° P·P°PIPµC°C°PµPSP°Cµ PiC°P°S
                                PiC°P°PjPjC<
000E CD 21                    int 21h                                ;
P·P°P
                                IPµC°C°PµPSP°Pµ PiC°P°PiC°P°PjPjC< P° PIC<C...PsP r° 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

1-1

Page

```

; HELLO2 - PJC†PµP±PSP°CŬ PİCŦPsPİCŦP°PjPjP°
N2      P»P°P±.CŦP°P±.#1 PİPs PŦPĖCŦC†PĖPİP»PĖPSPµ
"P      ħCŦC...PĖC,PµPĖC,CŦCŦP° PĖPsPjPİCŦCŦC,PµCŦP°"
;      PŬCŦPsPİCŦP°PjPjP°
PĖCŦPİPsP»CŦP·CŦP

```

```

CfC,                                µC, PīCtPSc†PµPŕCŕCtCŕ PŕP»Cµ PīPµC†P°C,Pë
                                     CtPSPePë
                                     ;
                                     ; PŷP•PbPŶPŷ PµP PŕP"P PŕPbPbP«

= 0024                                EOFLine EQU '$' ;
PŕPīCtPµPŕPµP»PµPSP                ëPµ CŕPëPjPIPS»CtPSPSPN PePSPSCŕC,P°PSC,C<
                                     ; "PbPSPSPµC†

CfC                                    ,CtPSPePë"

                                     ; PŶC,PµPe PīCtPSPiCtP°PjPjC<

ASSUME CS:CODE, SS:AStack

0000                                AStack SEGMENT STACK
0000 000C[                            DW 12 DUP('!') ;
PŕC,PIPSŕPëC,CŕC                    µ 12 CŕP»PSPI PīP°PjCµC,Pë
0021                                ]

0018                                AStack ENDS

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

0000                                DATA SEGMENT

                                     ; P"PëCtPµPeC,PëPIC< PSPIPëCŕP°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 PīCtPSPiCtP°PjPjC<

0000                                CODE SEGMENT
                                     ; PµCtPSc†PµPŕCŕCtP° PīPµC†P°C,Pë CfC,CtPSPePë
WriteMsg PROC NEAR
0000 B4 09                            mov AH,9
0002 CD 21                            int 21h ; P'C<P•PSPI

C,,CŕPSPeC†PëP                      ë DOS PīPs PīCtPµCtC<PIP°PSPëCt

0004 C3                                ret
0005                                WriteMsg ENDP

                                     ; P"PSP»PSPIPSP°Cµ PīCtPSc†PµPŕCŕCtP°

```

1-2

Address	Disassembly	Comment
0005	1E	Main PROC FAR
0005	push DS ; \	
0006	2B C0	Pμ P°PᵣCᵇPμCᵣP° PSP°C‡P°P»P° PSP PI CᵣC, PμPePμ
0006	sub AX, AX ; > PᵣP»Cᵇ	
0008	50	rᵣCᵣCᵇC%PμPᵢPs PᵢPsCᵣCᵣC, P°PSPsPᵢP»PμPSPëCᵇ
0008	push AX ; / PePsPjP°PSPᵣPμ	
0009	B8 ---- R	, P·P°PᵢPμCᵇC€P°CᵇC%PμPᵢPᵇ PᵢCᵇPsC‡PμPᵣCᵣCᵇCᵣ.
0009	mov AX, DATA ; P—	
000C	8E D8	ᵣP·PeP° CᵣPμPᵢPjPμPSC, PSPsPᵢPs
000C	mov DS, AX ;	
000E	BA 0000 R	ᵣC, CᵇP° PᵣP°PSPSC<C...
000E	mov DX, OFFSET HELLO ;	
0011	E8 0000 R	ᵣ PSP° CᵣPeCᵇP°PS PᵢPμCᵇPᵢPsPᵇ
0011	call WriteMsg ;	
0014	BA 0010 R	ëPë PᵢCᵇPëPᵢPμC, CᵣC, PᵢPëCᵇ.
0014	mov DX, OFFSET GREETING ;	
0017	E8 0000 R	ᵣ PSP° CᵣPeCᵇP°PS PᵢC, PsCᵇPsPᵇ
0017	call WriteMsg ;	
001A	CB	ëPë PᵢCᵇPëPᵢPμC, CᵣC, PᵢPëCᵇ.
001A	ret ;	
001B		ᵣ PI DOS PᵢPs PePsPjP°PSPᵣPμ,
001B		;
001B		PSP°C...PsP
001B		rᵣCᵇC%PμPᵇCᵣCᵇ PI 1-PsPj CᵣP»PsPᵢPμ PSP.
001B		Main ENDP
001B		CODE ENDS
001B		END Main

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