

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: Трансляции, отладка и выполнение программ на языке
Ассемблера.

Студент гр. 1381

Новак П.И.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

Развитие навыков работы с языком программирования Ассемблер, таких как: вызов трансляции, отладки и запуск программы.

Задание.

Лабораторная работа 1 использует 2 готовых программы на ассемблере: `hello1` – составлена с использованием сокращенного описания сегментов и `hello2` – составлена с полным описанием сегментов и выводом строки, оформленным как процедура. Выполнение работы состоит из двух частей, по каждой из которых необходимо представить протокол с фиксацией всех выполняемых действий и полученных результатов, и подписать его у преподавателя.

Ход работы.

1. Ознакомиться с программой `hello1.asm`, которая формирует и выводит на экран приветствие пользователя с помощью функции ОС MSDOS, вызываемой через прерывание с номером 21H (команда `Int 21h`).

2. Разобраться в структуре и реализации каждого сегмента программы. Непонятные фрагменты прояснить у преподавателя. Строку-приветствие преобразовать в соответствии со своими личными данными.

3. Загружен файл `hello1.asm` из каталога Задания в каталог Masm.

4. Протранслирована программа с помощью строки `> masm hello1.asm` с созданием объектного файла и файла диагностических сообщений (файла листинга).

5. Скомпонован загрузочный модуль с помощью строки `> link hello1.obj` с созданием карты памяти и исполняемого файла `hello1.exe`.

6. Выполнена программа в автоматическом режиме путем набора строки `> hello1.exe`

7. Запущено выполнение программы под управлением отладчика с помощью команды `> afdpro hello1.ex`

hello1.asm

(CS)=1A05

(DS)=19F5

(ES)=19F5

(SS)=1A0C

Адрес команды	Символический код команды	16-ричный код команды	Изменяемые данные	
			до	после
0010	MOV AX, 1A07	B8071A	AX = 0000 IP = 0010	AX = 1A07 IP = 0013
0013	MOV DS, AX	8ED8	DS = 19F5 IP = 0013 AX = 1A07	DS = 1A07 IP = 0015 AX = 1A07
0015	MOV DX, 0000	BA0000	IP = 0015 DX = 0000	IP = 0018 DX = 0000
0018	MOV AH, 09	B409	AX = 1A07 IP = 0018	AX = 0907 IP = 001A
001A	INT 21	CD21	IP = 001A	F000 IP = 001C
001C	MOV AH, 4C	B44C	AX = 0907 IP = 001C	AX = 4C07 IP = 001E
001E	INT 21	CD21	AX = 4C07 DS = 1A07 IP = 001E	AX = 0000 DS = 19F5 IP = 0010

hello2.asm

(CS)=1A0A

(DS)=19F5

(ES)=19F5

(SS)=1A05

Адрес команды	Символический код команды	16-ричный код команды	Изменяемые данные	
			до	после
0005	PUSH DS	1E	SP = 0018 IP = 0005 DS = 19F5 STACK(+0) = 0000	SP = 0016 IP = 0006 DS = 19F5 STACK(+0) = =19F5
0006	SUB AX,AX	2BC0	IP = 0006	IP = 0008
0008	PUSH AX	50	SP = 0016 IP = 0008 AX = 0000 STACK(+0) = 19F5 STACK(+2) = 0000	SP = 0014 IP = 0009 AX = 0000 STACK(+0) = 0000 STACK(+2) = 19F5
0009	MOV AX,1A07	B8071A	IP = 0009 AX = 0000	IP = 000C AX = 1A07
000C	MOV DS,AX	8ED8	DS = 19F5 IP = 000C AX = 1A07	DS = 1A07 IP = 000E AX = 1A07
000E	MOV DX,0000	BA0000	IP = 000E DX = 0000	IP = 0011 DX = 0000
0011	CALL 0000	E9ECFF	SP = 0014 IP = 0011 STACK(+0) =	SP = 0012 IP = 0000 STACK(+0) =

			0000 STACK(+2) = 19F5 STACK (+4) = 0000	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	3836FF	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	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	CS = 1A0A SP = 0014 IP = 001A STACK(+2) = 19F5	CS = 19F5 SP = 0018 IP = 0000 STACK(+2) = 0000
0000	INT 20	CD20	AX = 0907 DX = 0010 CS = 19F5 DS = 1A07 IP = 0000	AX = 0000 DX = 0000 CS = 1A0A DS = 19F5 IP = 0005

Выводы.

В ходе выполнения лабораторной работы были получены основные навыки программирования на ассемблере.

Приложение А. Код программ.

Имя файла: hello1.asm

DOSSEG

.MODEL SMALL

.STACK 100h

.DATA

Greeting LABEL BYTE

DB 'Вас приветствует уч.гр.1381 - Новак П.И.',13,10,'\$'

.CODE

mov ax, @data

mov ds, ax

mov dx, OFFSET Greeting

DisplayGreeting:

mov ah, 9

int 21h

mov ah, 4ch

int 21h

END

Имя файла: 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 – Novak Polina$'
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
```

```
Main
```


; (PSPsPjPμCῼ 09 PīCῼPμCῼCϵPIP°PSPē

Pμ 21h), PēPsC,PsCῼP°Cῼ:

; - PsP±PμCῼPīPμCῼPēPIP°PμC, PICϵP

IPsPr PSP° CῼPēCῼP°PS CῼC,CῼPsPēPē CῼPēPjPIP sP»

PsPI,

; P·P°PēP°PSCῼPēPIP°CῼC%PμPNῼCῼCῼ

P·PSP°PēPsPj "\$";

; - C,CῼPμP±CῼPμC, P·P°PrP°PSPēCῼ

PI CῼPμPiPēCῼC,CῼPμ ah PSPsPjPμCῼP° C,,CῼPSPēCῼP

ēPē=09h,

; P° PI CῼPμPiPēCῼC,CῼPμ dx - C

ῼPjPμC%PμPSPēCῼ P°PrCῼPμCῼP° PICϵPIP sPrPēPjPs

PNῼ

; CῼC,CῼPsPēPē;

; - PēCῼPīPsP»CῼP·CῼPμC, CῼPμPiPēC

ῼC,Cῼ ax Pē PSPμ CῼPsC...CῼP°PSCῼPμC, PμPiPs

; CῼPsPrPμCῼPῼPēPjPsPμ.

; *****

DOSSEG

; P—P°PrP°PSPēPμ CῼPμPiPjPμPSC,PsPI PīPsPr P”Ph

PŸ

.MODEL SMALL

; P_hP_sP_rP_μP»C_h P_iP°P_jC_lC,Pë-SMALL(P_hP°P»P°C_l)

.STACK 100h

; P_hC,P_iP_μC_lC,Pë P_iP_sP_r PŸC,P_μP_e 256 P±P°P_N2C,

.DATA

; P_kP°C‡P°P»P_s C_lP_μP_iP_jP_μP_sC,P° P_rP°P_sP_sC<C...

0000 Greeting LABEL BYTE

; PŸP_μP_eC_lC, P_iC_hPëP_iP_μC,C_lC,P_iPëC_l

0000 D0 92 D0 B0 D1 81 DB 'P'P°C_l P_iC_hPëP_iP_μC,C_lC,P_iC_lP_μC,
C_lC‡.P_iC

h.1381 - P_kP_sP_iP°P_e P_μ.P□.',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 83 D1 87

2E D0 B3 D1 80 2E

31 33 38 31 20 2D

20 D0 9D D0 BE D0

B2 D0 B0 D0 BA 20

D0 9F 2E D0 98 2E

0D 0A 24

.CODE ; PќP°C

‡P°P»Ps CЃPμPiPjPμPSC,P° PєPsPrP°

0000 B8 ---- R mov ax, @data ; P—P°

PiCЃCЃP·PєP° PI DS P°PrCЃPμCЃP° PSP°C‡P°P»P°

0003 8E D8 mov ds, ax ; CЃPμ

PiPjPμPSC,P° PrP°PSPSC<C...

0005 BA 0000 R mov dx, OFFSET Greeting ; P—P°

PiCЃCЃP·PєP° PI dx CЃPjPμC%oPμPSPëCЃ

; P°PrC

ЃPμCЃP° C,PμPєCЃC,P° PїCЃPëPIPμC,CЃC,PIPëCЃ

0008 DisplayGreeting:

0008 B4 09 mov ah, 9 ; # C,,

CЃPSPєC†PëPë P”PhPŸ PiPμC‡P°C,Pë CЃC,CЃPsPєPë

000A CD 21 int 21h ; PIC<P

IPsPr PSP° C'PeCTP°PS PiCTPëPIPμC,C'IC,PIPëC΀

000C B4 4C mov ah, 4ch ; # C,,

C'PSPeC†PëPë P''PhPŸ P·P°PIPμCTC€PμPSPëC΀

PiCTPs

PiCTP°PjPjC<

000E CD 21 int 21h ; P·P°P

IPμCTC€PμPSPëPμ PiCTPsPiCTP°PjPjC< Pë PIC<C...PsP

r PI P''PhPŸ

END

Symbols-1

Segments and Groups:

N a m e	Length	Align	Combine	Class
DGROUP	GROUP			
_DATA	0045	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

Имя файла: hello2.lst

Microsoft (R) Macro Assembler Version 5.10

9/21/22 22:39:06

Page 1-1

; HELLO2 - PJC‡PμP±PSP°CЦ PιCЪPsPiCЪP°PjPjP° N2

P»P°P±.CЪP°P±.#1 PiPs PrPёCГC†PёPiP»PёPSPμ "P

ђCЪC...PёC,PμPёC,CГCЪP° PёPsPjPiCЪCЪC,PμCЪP°"

```

;      PuChPsPiChPoPjPjPo PeClPiPsP»ChP·CfP
μC, PiChPsC†PμPrC†ChCf PrP»Cl PiPμC‡PoC, Pe ClC,
ChPsPePe
;
;      PŷP·PлPŸPŷ PuP PhP“P PhPhPhPhP«

```

```

= 0024      EOFLine EQU '$'      ; PhPiChPμPrPμP»PμPSP
Pμ ClPePjPiPsP»ChPSPsPN PePsPSClC, PoPSC, Cх
;      "PлPsPSPμC† ClC
,ChPsPePe"

```

```

; PŸC, PμPe PiChPsPiChPoPjPjCх

```

```

ASSUME CS:CODE, SS:AStack

```

```

0000      AStack SEGMENT STACK
0000 000C[      DW 12 DUP('!') ; PhC, PiPsPrPeC, ClC
Ц 12 ClP»PsPi PiPoPjClC, Pe
0021
]

```

```

0018      AStack ENDS

```



```
; P”P°PSPSC<Pμ PïCṪPsPiCṪP°PjPjC<
```

```
0000          DATA    SEGMENT
```

```
; P”PëCṪPμPεC,PëPIC< PsPïPëCÍP°PSPëCΠ 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 4350 - $'
      74 20 66 72 6F 6D
      20 34 33 35 30 20
      2D 20 24
```

```
0025          DATA    ENDS
```

```
; PЉPsPr PïCṪPsPiCṪP°PjPjC<
```

```
0000          CODE     SEGMENT
```

```
; PұCṪPsC†PμPrCfCṪP° PïPμC‡P°C,Pë CÍC,CṪPsPePë
```

```
0000          WriteMsg PROC NEAR
```

```

0000 B4 09          mov  AH,9
0002 CD 21          int  21h ; P'C<P·PsPI C,,CfPSPeC†PëP
                        ë DOS PïPs PïCᵀPμCᵀC<PIP°PSPëCᵀ
0004 C3            ret
0005              WriteMsg ENDP

```

```

; P“PsP»PsPIPSP°Cᵀ PïCᵀPsC†PμPrCfCᵀP°

```

```

0005                      Main    PROC FAR

0005 1E                      push DS      ;\ PŸPsC...CтP°PSPμPSPë
                                Pμ P°PrCтPμCГP° PSP°C‡P°P»P° PSP PI CГC,PμPεPμ
                                rCГCтC%oPμPiPs      PIPsCГCГC,P°PSPsPIP»PμPSPëCЦ
0006 2B C0                  sub  AX,AX    ; > PrP»CЦ PiPsCГP»PμP
                                rCГCтC%oPμPiPs      PIPsCГCГC,P°PSPsPIP»PμPSPëCЦ
PiPs
0008 50                      push AX      ;/ PεPsPjP°PSPPrPμ ret
                                , P·P°PIPμCтC€P°CтC%oPμPN₂ PiCтPsC‡PμPrCГCтCГ.
0009 B8 ---- R              mov  AX,DATA      ; P—P°PiCтC
                                íP·PεP° CГPμPiPjPμPSC,PSPsPiPs
000C 8E D8                  mov  DS,AX        ; CтPμPiPëC
                                ГC,CтP° PrP°PSPSC<C....
000E BA 0000 R              mov  DX, OFFSET HELLO ; P'C<PIPsP
                                r PSP° CKPεCтP°PS PiPμCтPIPsPN₂
0011 E8 0000 R              call WriteMsg      ; CГC,CтPsP
                                εPë PiCтPëPIPμC,CГC,PIPëCЦ.
0014 BA 0010 R              mov  DX, OFFSET GREETING ; P'C<PIPsP
                                r PSP° CKPεCтP°PS PIC,PsCтPsPN₂
0017 E8 0000 R              call WriteMsg      ; CГC,CтPsP

```

сРё РїСѢРёРІРµС,СЃС,РІРёСЏ.

001A CB ret ; Р'С‹С...РѕР

р РІ DOS РїРѕ РёРѕРјР°РѕРѕРјРµ,

; РѕР°С...РѕР

рСЏС%РµРµСЃСЃ РІ 1-РѕРј СЃР»»РѕРІРµ РѕР.

001B Main ENDP

001B CODE ENDS

END Main

Symbols-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

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