

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: Изучение трансляции, отладки и выполнения программ на
языке Ассемблера

Студент(ка) гр. 9382

Голубева В.П.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

$\neg \hat{A} \otimes \tilde{A} \odot \tilde{A} \neg \tilde{A} \otimes \quad \tilde{A} \in \tilde{A} \varepsilon \hat{A} \hat{A}^a \tilde{A} \check{S} \hat{A}_i \hat{A}_i; \quad \hat{A} \langle \hat{A} \quad \hat{A}$
 $"\hat{A}(\tilde{A} \ll \hat{A} \wp \hat{A} \otimes \hat{a}(\neg \tilde{A}_i$
 $\tilde{A} \wp \tilde{A} \quad \hat{A} \otimes \hat{A}^a \hat{A}_i;"$
 $;$ $(\hat{A} \hat{A} \otimes \hat{A} \neg \hat{A} \forall \tilde{A} \quad 09 \hat{A}^- \tilde{A} \hat{A} \forall \tilde{A} \quad \tilde{A} \ll \hat{A} \wp \hat{A} \quad \hat{A} \hat{A}_i$
 $\hat{A} \forall 21h), \hat{A}^a \hat{A} \otimes \tilde{A} \wp \hat{A} \otimes \tilde{A} \quad \hat{A} \quad \tilde{A}^-:$
 $;$ $- \hat{A} \otimes \hat{A}_i \hat{A} \forall \tilde{A}_i \hat{A}^- \hat{A} \forall \tilde{A} \check{S} \hat{A}_i \hat{A} \wp \hat{A} \quad \hat{A} \forall \tilde{A} \wp \quad \hat{A}$
 $\wp \tilde{A} \ll \hat{A}$
 $\wp \hat{A} \otimes \hat{a}(\neg \quad \hat{A} \hat{A} \quad \tilde{A} \hat{A}^a \tilde{A} \quad \hat{A} \quad \tilde{A}_i \tilde{A}$
 $\wp \tilde{A} \quad \hat{A} \otimes \hat{A}^a \hat{A}_i \quad \tilde{A}_i \hat{A}_i \hat{A} \neg \hat{A} \wp \hat{A} \otimes \hat{A}$
 $\ll \hat{A} \otimes \hat{A} \wp,$
 $;$ $\hat{A} \check{S} \hat{A} \quad \hat{A}^a \hat{A} \quad \hat{A} \tilde{A} \check{S} \hat{A}_i \hat{A}$
 $\wp \hat{A} \quad \tilde{A} \otimes \tilde{A} \odot \hat{A} \forall \hat{A} \odot \tilde{A}_i \tilde{A}^-$
 $\hat{A} \check{S} \hat{A} \hat{A} \quad \hat{A}^a \hat{A} \otimes \hat{A} \neg "S";$
 $;$ $- \tilde{A} \wp \tilde{A} \quad \hat{A} \forall \hat{A}_i \tilde{A} \varepsilon \hat{A} \forall \tilde{A} \wp \quad \hat{A} \check{S} \hat{A} \quad \hat{a}(\neg \hat{A} \quad \hat{A} \hat{A}_i \tilde{A}^-$
 $\hat{A} \wp \quad \tilde{A} \quad \hat{A} \forall \hat{A} \varepsilon \hat{A}_i \tilde{A}_i \tilde{A} \wp \tilde{A} \quad \hat{A} \forall \quad ah \quad \hat{A} \hat{A} \otimes \hat{A} \neg \hat{A} \forall \tilde{A} \quad \hat{A}$
 $\tilde{A} \in \tilde{A} \varepsilon \hat{A} \hat{A}^a \tilde{A} \check{S}$
 $\hat{A}_i \hat{A}_i = 09h,$
 $;$ $\hat{A} \quad \hat{A} \wp \quad \tilde{A} \quad \hat{A} \forall \hat{A} \varepsilon \hat{A}_i \tilde{A}_i \tilde{A} \wp \tilde{A} \quad \hat{A} \forall \quad dx - \quad \tilde{A}$
 $_i \hat{A} \neg \hat{A} \forall \tilde{A} \odot \hat{A} \forall \hat{A} \hat{A}_i \tilde{A}^- \quad \hat{A} \quad \hat{a}(\neg \tilde{A} \quad \hat{A} \forall \tilde{A}_i \hat{A} \quad \hat{A} \wp \tilde{A} \ll \hat{A}$
 $\wp \hat{A} \otimes \hat{a}(\neg \hat{A}_i \hat{A} \neg$
 $\hat{A} \otimes \hat{A} \odot$
 $;$ $\tilde{A}_i \tilde{A} \wp \tilde{A} \quad \hat{A} \otimes \hat{A}^a \hat{A}_i;$
 $;$ $- \hat{A}_i \tilde{A}_i \hat{A}^- \hat{A} \otimes \hat{A} \ll \tilde{A} \neg \hat{A} \check{S} \tilde{A} \varepsilon \hat{A} \forall \tilde{A} \wp$
 $\tilde{A} \quad \hat{A} \forall \hat{A} \varepsilon \hat{A}_i \tilde{A}$
 $_i \tilde{A} \wp \tilde{A} \quad ax \quad \hat{A}_i \quad \hat{A} \hat{A} \forall \quad \tilde{A}_i \hat{A} \otimes \tilde{A} \forall \tilde{A} \quad \hat{A} \quad \hat{A} \tilde{A}^- \hat{A} \forall \tilde{A} \wp$
 $\hat{A} \forall \hat{A} \varepsilon \hat{A} \otimes$
 $;$ $\tilde{A}_i \hat{A} \otimes \hat{a}(\neg \hat{A} \forall \tilde{A} \quad \hat{A} \quad \hat{A}_i \hat{A} \neg \hat{A} \otimes \hat{A} \forall.$

;

DOSSEG

; Â»Â â(¬Â ÂĀ;Â¥ Â;Â¥Â£Â¬Â¥ÂÃ¢Â®Â¢

Â¬Â®â(¬Â(

Â Â

.MODEL SMALL

; Â{Â®â(¬Â¥Â«Â¬ Â¬Â Â¬Â¬Â¢Ā;-

SMALL(Â{Â Â«Â Â¬)

.STACK 100h

; Â Â¢Â¢Â¥Ā;Ā¢Ā; Â¬Â®â ¬Â Â¢Â¥Âª 256

Ā;Â Â©Ā¢

.DATA

; Â Â Ā§Â Â«Â® Ā;Â¥Â£Â¬Â¥ÂÃ¢Â â(¬Â Â-

ÂĀ«Ā¥

0000

Greeting LABEL BYTE

; Â Â¥ÂªĀ;Ā¢ Â¬Ā Ā;Â¢Â¥Ā¢Ā;Ā¢Â¢Ā;Ā¬

0000 68 65 6C 6C 6F 20

DB 'hello 1.',13,10','\$'

31 2E 0D 0A 24

.CODE

; Â Â Ā

§Â Â«Â® Ā;Â¥Â£Â¬Â¥ÂÃ¢Â ÂªÂ®â(¬Â

0000 B8 ---- R

mov ax, @data

; Â»Â

Â£Ā Ā£Â§ÂªÂ Â¢ DS Â â(¬Ā Â¥Ā;Â Â-

Â Ā§Â Â«Â

0003 8E D8

mov ds, ax

; Ā;Â¥

Â£Â¬Â¥ĀÃ¢Â â(¬Â ÂĀĀ«Ā¥

```
0005 BA 0000 R      mov dx, OFFSET Greeting      ; Â»Â
```

```

        Â£Ã  Ã£Â§ÂªÂ  Â¢ dx  Ã;Â¬Â¥Ã©Â¥Â;Ã¬
                                ; Â â(¬
        Ã  Â¥Ã;Â  Ã¢Â¥ÂªÃ;Ã¢Â  Â¬Ã  Â;Â¢Â¥Ã¢Ã;Ã
        ¢Â¢Â;Ã¬
0008      DisplayGreeting:
0008  B4 09      mov  ah, 9                ; # Ã€
        Ã£ÂªÂªÃŒÂ;  Â(Â  Â  Â¬Â¥Ã§Â  Ã¢Â;  Ã;Ã
        ¢Ã  Â®ÂªÂ;
000A  CD 21      int  21h                ; Â¢Ã«Â
        ¢Â®â(¬  ÂÂ  ÃªÃ  Â  Â  Â¬Ã  Â;Â¢Â¥Ã¢Ã;Ã¢Â
        ¢Â;Ã¬
000C  B4 4C      mov  ah, 4ch            ; # Ã€
        Ã£ÂªÂªÃŒÂ;  Â(Â  Â  Â§Â  Â¢Â¥Ã  ÃŒÂ¥Â¬
        Â;Ã¬  Â¬Ã  Â®
        Â£Ã  Â  Â¬Â¬Ã«
000E  CD 21      int  21h                ; Â§Â  Â
        ¢Â¥Ã  ÃŒÂ¥ÂªÂ;Â¥  Â¬Ã  Â®Â£Ã  Â  Â¬Â¬Ã«
        Â;  Â¢Ã«Ã¥Â®â
        (¬  Â¢  Â(Â  Â
        END

```

Symbols-1

Segments and Groups:

N a m e	Length	Align	Combine	Class
DGROUP	GROUP			
_DATA	000B	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

47992 + 432676 Bytes symbol space free

0 Warning Errors

0 Severe Errors

1) Создаём логический диск «с» с помощью команды `> mount c home/i/tools`, задающий путь к `MASM.EXE`. Переходим в диск командой `>c:`

2) Открываем программу `HELLO1.ASM` в текстовом редакторе, заменяем строку приветствие на «hello 1»

3) Транслируем команду с помощью `> masm HELLO1.ASM`, создаем объектный файл `HELLO1.OBJ` `NUL.LST`. Синтаксических ошибок обнаружено не было. Результат представлен в рисунке 1



```
C:\>masm HELLO1.ASM
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.

Object filename [HELLO1.OBJ]:
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:

50050 + 432668 Bytes symbol space free

0 Warning Errors
0 Severe Errors
```

Рисунок 1.

4) Компилируем загрузочный модуль `>link HELLO1.OBJ`. Создался загрузочный модуль `HELLO1.EXE`. Результат представлен в рисунке 2


```

C:\>link HELLO1.OBJ
Packed file is corrupt
Now run it with LOADFIX..

Microsoft (R) Overlay Linker Version 3.64
Copyright (C) Microsoft Corp 1983-1988. All rights reserved.

Run File [HELLO1.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:

```

Рисунок 2.

5) Выполним программу с помощью команды >HELLO1.EXE. Результат представлен в рисунке 3.

```

C:\>HELLO1.EXE
hello 1.

```

Рисунок 3.

6) Выполним программу в пошаговом режиме с помощью команды >AFDPRO.EXE HELLO1.EXE. Обычные команды выполняются по F1, а вызовы обработчиков прерывания (Int) - по F2, чтобы не входить внутрь обработчика прерываний.

Результаты пошагового выполнения программы:

Начальные данные представлены в рисунке 4

```

AX 0000 SI 0000 CS 30C0 IP 0010 Stack +0 0000
BX 0000 DI 0000 DS 30B0      +2 0000
CX 0067 BP 0000 ES 30B0 HS 30B0 +4 0000
DX 0000 SP 0100 SS 30C7 FS 30B0 +6 0000

```

Рисунок 4.

Процесс пошагового выполнения в таблице1:

Таблица 1

Адрес команды	Шестнадцатер ичный код команды	Символьный код команды	Содержимое в ячейках памяти (до выполнения/после
------------------	--------------------------------------	---------------------------	--

			выполнения)	
0010	B8C230	MOV AX, 30C2	IP=0010 AX=0000	IP=0013 AX=30C2
0013	8ED8	MOV DS, AX	IP=0013 AX=30C2 DS=30B0	IP=0015 AX=30C2 DS=30C2
0015	BA0000	MOV DX, 0000	IP=0015 AX=30C2 DS=30C2	IP=0018 AX=30C2 DS=30B0
0018	B409	MOV AH, 09	IP=0018 AX=30C2	IP=001A AX=09C2
001A	CD21	INT 21	IP=001A AX=09C2	IP=001C AX=0924
001C	B44C	MOV AH, 4C	IP=001C AX=0924	IP=001E AX=4C24
001E	CD21	INT 21	IP=001E AX=4C24	IP=0010 AX=0000

ПРИЛОЖЕНИЕ 2

HELLO2.LST

```

; HELLO2 -  çŸ; i ¯à®£à ¬¬ N2 « ;.à ;.#1 ¯®
€šáæš ¯«šŸ "  ¯àâšâŸªââà ¯®¬ ¯îâŸà "
;      à®£à ¬¬ šá ¯®«ì§âŸâ ¯à®æŸ€ãâã €«i ¯
Ÿç âš áâà®ªš
;
;      )(      )(H

= 0024      EOFLine EQU '$'      ; ¯àŸ€Ÿ«ŸšŸ áš¬
¢®«ì

®© ¯®áâ âë
;      "(®Ÿæ áâà®ªš"

;  âŸª ¯à®£à ¬¬ë

0000      AStack  SEGMENT STACK
0000 000C[      DW 12 DUP(?)      ;  â¢®€šâî 12 á«®
¢ ¯

      ¬îâš

      ????

      ]

0018      AStack  ENDS

; ( ëŸ ¯à®£à ¬¬ë

```

0000 DATA SEGMENT

; (šà¥ªâš¢ë ®¯śá šĩ € ěå

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 9382 - \$'

74 20 66 72 6F 6D

20 39 33 38 32 20

2D 20 24

0025 DATA ENDS

; (®€ ¯à®£à ¯¬ë

0000 CODE SEGMENT

ASSUME CS:Code DS:DATA SS:AStack

HELLO2.ASM(28): warning A4001: Extra characters on line

; à®æ¥£ää ¯¥ç âš áâà®ªš

0000 WriteMsg PROC NEAR

0000 B4 09 mov AH,9

0002 CD 21 int 21h ; (ë§®¢ äãªæšš DOS ¯® ¯à¥

àë¢ šî

0004 C3 ret

0005 WriteMsg ENDP

```

; @«@¢ ¯à@æ¥€ãà
0005 Main PROC FAR
0005 1E push DS ;\ @åà ¥š¥ €à¥á
ç « PSP ¢ áâ¥ª¥
0006 2B C0 sub AX,AX ; > €«ï ¯@á«¥€ãîé¥£®
¢
®áâ® ®¢«¥šï ¯®
#Microsoft (R) Macro Assembler Version 5.10 10/1/20 16:11:45
Page 1-2

0008 50 push AX ;/ ª®¬ €¥ ret, § ¢¥à
è îé¥© ¯à@æ¥€ãàã.
0009 B8 ---- R mov AX,DATA ; › £ãã§ª
á¥£¬¥â®£®
000C 8E D8 mov DS,AX ; à¥£šáâ
€ äå.
000E BA 0000 R mov DX, OFFSET HELLO ; (ë¢®€
íªà ¯¥à¢®©
0011 E8 0000 R call WriteMsg ; áâà®ªš ¯à
š¢¥âââ¢ši.
0014 BA 0010 R mov DX, OFFSET GREETING ; (ë¢®€
íªà ¢â®à®©
0017 E8 0000 R call WriteMsg ; áâà®ªš ¯à
š¢¥âââ¢ši.
001A CB ret ; (ëå®€ ¢ D
OS ¯® ª®¬ €¥,
; å®€îé¥©

```

Microsoft (R) Macro Assembler Version 5.10

```
001B      Main      ENDP
001B      CODE      ENDS
                END Main
```

10/1/20 16:11:45

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

51 Source Lines

51 Total Lines

13 Symbols

47996 + 434719 Bytes symbol space free

1 Warning Errors

0 Severe Errors

1) Открываем программу HELLO2.ASM в текстовом редакторе, заменяем строку приветствие

2) Транслируем команду с помощью > masm HELLO2.ASM, создаем объектный файл HELLO2.OBJ, а также NUL.LST. Результат представлен в рисунке 5

```

C:\>masm HELLO2.ASM
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.

Object filename [HELLO2.OBJ]:
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:
HELLO2.ASM(28): warning A4001: Extra characters on line

50058 + 434707 Bytes symbol space free

1 Warning Errors
0 Severe Errors

```

Рисунок 5.

3) Компилируем загрузочный модуль >link HELLO12OBJ. Создался загрузочный модуль HELLO2.EXE. Результат представлен в рисунке 6

```

C:\>link HELLO2.OBJ
Packed file is corrupt
Now run it with LOADFIX..

Microsoft (R) Overlay Linker Version 3.64
Copyright (C) Microsoft Corp 1983-1988. All rights reserved.

Run File [HELLO2.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:

```

Рисунок 6

4) Выполним программу с помощью команды >HELLO2.EXE. Результат представлен в рисунке 7

```

C:\>HELLO2.EXE
Hello Worlds!
Student from 9382 -

```

Рисунок 7.

5) Выполним программу в пошаговом режиме с помощью команды >AFDPRO.EXE HELLO2.EXE. Обычные команды выполняются по F1, а вызовы обработчиков прерывания (Int) - по F2, чтобы не входить внутрь обработчика прерываний. Результат представлен в рисунке 8


```

AX 0000 SI 0000 CS 30C5 IP 0005 Stack +0 0000
BX 0000 DI 0000 DS 30B0      +2 0000
CX 006B BP 0000 ES 30B0 HS 30B0      +4 0000
DX 0000 SP 0018 SS 30C0 FS 30B0      +6 0000

```

Рисунок 8

Результаты пошагового выполнения программы в таблице2:

Таблица 2

Адрес команды	Шестнадцатер ичный код команды	Символьный код команды	Содержимое в ячейках памяти (до выполнения/после выполнения)	
0005	1E	PUSH DS	IP=0005 AX=0000 STACK +0 0000 SP=0018	IP=0006 STACK +0 30B0 SP=0016
0006	2BC0	SUB AX, AX	IP=0006 STACK +0 30B0 SP=0016	IP=0008
0008	50	PUSH AX	IP=0008 SP=0016	IP=0009 STACK +2 30B0 SP=0014
0009	B8C230	MOV AX, 30C2	IP=0009 STACK +2 30B0 SP=0014	IP=000C AX=30C2

000C	8ED8	MOV DS, AX	IP=000C AX=30C2	IP=000E
000E	BA0000	MOV DX, 0000	IP=000E	IP=0011
0011	E8ECFF	CALL 0000	IP=0011 SP=0014	IP=0000 STACK +0 0014 +4 30B0 SP=0012
0000	B409	AH, 09	IP=0000 STACK +0 0014 +4 30B0	IP=0002 AX=09C2
0002	CD21	INT 21	IP=0002 AX=09C2	IP=0004 AX=0924
0004	C3	RET	IP=0004 STACK +0 0014 +2 0000 +4 30B0	IP=0014 STACK +2 30B0 +4 0000
0014	BA1000	MOV DX, 0010	IP=0014 SP=0012 DX=0000	IP=0017 SP=0014 DX=0010
0017	E8E6FF	CALL, 0000	IP=0017 SP=0014	IP=0000 SP=0012 STACK +0 001A +4 030B0
0000	B409	MOV AH,	IP=0000	IP=0002

		09		
0002	CD 21	INT 21	IP=0002	IP=0004
0004	C3	RET	IP=0004 SP=0012 STACK +2 0000 +4 30B0	IP=001A SP=0014 STACK +2 30B0 +4 0000
001A	CB	RET	IP=001A SP=0014 AX=0924	IP=0000 SP=0018 AX=0000
0000	CD20	INT 20	IP=0000	IP=0005

Вывод

В процессе выполнения работы была изучена структура программ на языке Ассемблера.