

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. 0382

Куликов М.Д.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

Изучить трансляцию, отладку и выполнение программ на языке Ассемблера.

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

Записать начальное содержимое сегментных регистров CS, DS, ES и SS. Выполнить программу в пошаговом режиме с фиксацией используемых регистров и ячеек памяти до и после выполнения каждой команды. Обычные команды выполняются по F1 (Step), а вызовы обработчиков прерываний (Int) - по F2 (StepProc), чтобы не входить внутрь обработчика прерываний. Продвижение по сегментам экранной формы отладчика выполняется с помощью клавиш F7 – F10 (up, down, left, right). Перезапуск программы в отладчике выполняется клавишей F3 (Retrieve). Выход из отладчика - по команде Quit.

Результаты прогона программы под управлением отладчика должны быть представлены в виде, показанном на примере одной команды в табл.1, и подписаны преподавателем.

Часть 2

Выполнить пункты 1 - 7 части 1 настоящего задания применительно к программе hello2.asm, приведенной в каталоге Задания, которая выводит на экран приветствие пользователя с помощью процедуры WriteMsg, а также использует полное определение сегментов. Сравнить результаты прогона под управлением отладчика программ hello1 и hello2 и объяснить различия в размещении сегментов.

Выполнение работы

1. Часть 1

1.1. Просмотрен код программы hello1.asm и прочитаны комментарии, разобрано действие каждой строки.

1.2. В строку Greeting были добавлены личные данные.

1.3. Протранслирована программа hello1.asm с созданием объектного файла hello1.obj и файлом листинга hello1.lst ,ошибок не обнаружено.

1.4. Скомпонован загрузочный модуль с созданием исполняемого файла hello1.exe.

1.5. Запущена программа в автоматическом режиме. Программа выполнилась без ошибок, результат корректен.

1.6. Программа запущена под управлением отладчика.

Результаты отладки программы:

Начальное значение сегментных регистров:

CS = 1A05 ; DS = 19F5 ;

ES = 19F5 ; SS = 1A0C ;

Таблица 1 - результаты отладки программы hello1.exe

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

2. Часть 2

2.1. Прделаны аналогичные шаги 1.1-1.6 для программы hello2.asm.

В строку Greeting дописаны личные данные.

Результаты отладки программы :

Начальное значение сегментных регистров:

CS = 1A0B ; DS = 19F5 ;

ES = 19F5 ; SS = 1A05 ;

2.2. Таблица 2 — Результаты отлидки программы hello2.exe.

Адрес Команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			до выполнения	После выполнения
0005	PUSH DS	1E	IP = 0005 SP = 0018 Stack +0 0000 +2 0000 +4 0000 +6 0000	IP = 0006 SP = 0016 Stack +0 19F5 +2 0000 +4 0000 +6 0000
0006	SUB AX, AX	2B C0	IP = 0006	IP = 0008
0008	PUSH AX	50	IP = 0008 SP = 0016 Stack +0 19F5 +2 0000 +4 0000 +6 0000	IP = 0009 SP = 0014 Stack +0 0000 +2 19F5 +4 0000 +6 0000
0009	MOV AX, 1A07	B8071A	IP = 0009	IP = 000C

			AX = 0000	AX = 1A07
000C	MOV DS, AX	8ED8	IP = 000C DS = 19F5	IP = 000E DS = 1A07
000E	MOV DX, 0000	BA0000	IP = 000E	IP = 0011
0011	CALL 0000	E8ECFF	IP = 0011 SP = 0014 Stack +0 0000 +2 19F5 +4 0000 +6 0000	IP = 0000 SP = 0012 Stack +0 0014 +2 0000 +4 19F5 +6 0000
0000	MOV AH, 09	B409	IP = 0000 AX = 1A07	IP = 0002 AX = 0907
0002	INT 21	CD21	IP = 0002	IP = 0004
0004	RET	C3	IP = 0004 SP = 0012 Stack +0 0014 +2 0000 +4 19F5 +6 0000	IP = 0014 SP = 0014 Stack +0 0000 +2 19F5 +4 0000 +6 0000
0014	MOV DX,0010	BA0100	IP = 0014 DX = 0000	IP = 0017 DX = 0010
0017	CALL 0000	E8E6FF	IP = 0017 SP = 0014 Stack +0 0000 +2 19F5 +4 0000 +6 0000	IP = 0000 SP = 0012 Stack +0 001A +2 0000 +4 19F5 +6 0000
0000	MOV AH, 09	B409	IP = 0000	IP = 0002
0002	INT 21	CD21	IP = 0002	IP = 0004
0004	RET	C3	IP = 0004 SP = 0012	IP = 001A SP = 0014

			Stack +0 001A +2 0000 +4 19F5 +6 0000	Stack +0 0000 +2 19F5 +4 0000 +6 0000
001A	RET FAR	CB	IP = 001A SP = 0014 CS = 1A0A Stack +0 0000 +2 19F5 +4 0000 +6 0000	IP = 0000 SP = 0018 CS = 19F5 Stack +0 0000 +2 0000 +4 0000 +6 0000

Вывод.

Была изучена трансляция, отладка и выполнение программ на языке Ассемблера

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: hello1.asm

```
; HELLO1.ASM - упрощенная версия учебной программы лаб.раб. N1
;               по дисциплине "Архитектура компьютера"
; *****
; Назначение: Программа формирует и выводит на экран приветствие
;               пользователя с помощью функции ДОС "Вывод строки"
;               (номер 09 прерывание 21h), которая:
;               - обеспечивает вывод на экран строки символов,
;               заканчивающейся знаком "$";
;               - требует задания в регистре ah номера функции=09h,
;               а в регистре dx - смещения адреса выводимой
;               строки;
;               - использует регистр ax и не сохраняет его
;               содержимое.
; *****

        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.1383 - Куликов М.Д..',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
программы
        int  21h                             ; завершение программы и выход в
ДОС
        END
```

Название файла: hello2.asm

```
; HELLO2 - Учебная программа N2 лаб.раб.#1 по дисциплине "Архитектура
компьютера"
;               Программа использует процедуру для печати строки
;
;               ТЕКСТ ПРОГРАММЫ

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 1383 - Kulikov M.D. $'
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 Б

ЛИСТИНГИ

Название файла: hello1.lst

Microsoft (R) Macro Assembler Version 5.10

9/26/22 00:16:51

Page 1-1

DOSSEG

.MODEL SMALL

.STACK 100h

.DATA

0000

Greeting LABEL BYTE

0000 D0 92 D0 B0 D1 81

DB "Вас приветствует ст.гр.1383 - Куликов

М.Д.,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

31 33 38 33 20 2D

20 D0 9A D1 83 D0
BB D0 B8 D0 BA D0
BE D0 B2 20 D0 9C
2E D0 94 2E 2E 0D
0A 24

.CODE

0000 B8 ---- R mov ax, @data
0003 8E D8 mov ds, ax
0005 BA 0000 R mov dx, OFFSET Greeting

0008 DisplayGreeting:

0008 B4 09 mov ah, 9
000A CD 21 int 21h
000C B4 4C mov ah, 4ch
000E CD 21 int 21h

END

Symbols-1

Segments and Groups:

N a m e	Length	Align	Combine	Class
DGROUP		GROUP		
_DATA	004A	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

Название файла: hello2.lst

Microsoft (R) Macro Assembler Version 5.10

9/26/22 00:23:12

Page 1-1

```
; HELLO2 - PJC‡PµP±PSP°CЦ PïCЪPsPiCЪP°PjPjP° N2
P»P°P±.CЪP°P±.#1 PïPs PrPëCГC†PëPïP»PëPSPµ "P
ђCЪC...PëC,PµPëC,CГCЪP° PëPsPjPïCЪCЪC,PµCЪP°"
;      PцCЪPsPiCЪP°PjPjP° PëCГPïPsP»CЪP·CГP
µC, PïCЪPsC†PµPrCГCЪCГ PrP»CЦ PïPµC‡P°C,Pë
```

CГC,

```
CЪPsPëPë
```

```
;
```

```
; PŷP•PљPŸPŷ PцP PhP“P PhPњPњP«
```

= 0024

```
EOFLine EQU '$' ;
```

```
PhPïCЪPµPrPµP»PµPSP
```

```
ëPµ CГPëPjPIPsP»CЪPSPsPNё PëPsPSCГC,P°PSC,C<
;      "PљPsPSPµC† CГC
,CЪPsPëPë"
```

```
; PŸC,PµPë PïCЪPsPiCЪP°PjPjC<
```

```
ASSUME CS:CODE, SS:AStack
```

0000

```
AStack SEGMENT STACK
```

0000 000C[

```
DW 12 DUP('') ;
```

```
PhC,PIPsPrPëC,CГC
```

```
Ц 12 CГP»PsPI PïP°PjCЦC,Pë
```

0021

]

0018

AStack ENDS

; P”P°PSPSC<Pμ PïCᵀPsPiCᵀP°PjPjC<

0000

DATA SEGMENT

; P”PëCᵀPμPeC,PëPIC< PsPïPëCᵀP°PSPëCΠ PᵀP°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 1383 - Kulikov M.D.
\$'

74 20 66 72 6F 6D

20 31 33 38 33 20

2D 20 4B 75 6C 69

6B 6F 76 20 4D 2E

44 2E 20 24

0032

DATA ENDS

; PᵐPsPr PïCᵀPsPiCᵀP°PjPjC<

0000

CODE SEGMENT

; PᵀCᵀPsC†PμPrCᵀCᵀP° PïPμC†P°C,Pë CᵀC,CᵀPsPePë

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°CЀ PïCтPsC†PμPrCтCтP°

0005                                Main    PROC FAR
0005 1E                                push DS    ;\ PŸPsC...CтP°PSPμPSPë
                                Pμ P°PrCтPμCЀP° PSP°C‡P°P»P° PSP PI CЀC,PμPePμ
0006 2B C0                            sub  AX,AX    ; > PrP»CЀ
PïPsCЀP»PμP
                                rCтCтC%oPμPiPs PIPsCЀCЀC,P°PSPsPIP»PμPSPëCЀ
PïPs
0008 50                                push AX    ;/ PePsPjP°PSPPrPμ ret
                                , P·P°PIPμCтC€P°CтC%oPμPN№ PïCтPsC†PμPrCтCтCт.
0009 B8 ---- R                        mov  AX,DATA    ; P—P°PiCтC
fP·PeP° CЀPμPiPjPμPSC,PSPsPiPs
000C 8E D8                            mov  DS,AX    ; CтPμPiPëC
ЀC,CтP° PrP°PSPSC<C....
000E BA 0000 R                        mov  DX, OFFSET HELLO ; P’C<PIPsP
r PSP° CЀPeCтP°PS PïPμCтPIPsPN№
0011 E8 0000 R                        call WriteMsg    ; CЀC,CтPsP
ePë PïCтPëPIPμC,CЀC,PIPëCЀ.
0014 BA 0010 R                        mov  DX, OFFSET GREETING ; P’C<PIPsP
r PSP° CЀPeCтP°PS PIC,PsCтPsPN№
0017 E8 0000 R                        call WriteMsg    ; CЀC,CтPsP
ePë PïCтPëPIPμC,CЀC,PIPëCЀ.
001A CB                                ret            ; P’C<C...PsP
r PI DOS PïPs PePsPjP°PSPPrPμ,

```

; PSP°C...PsP

rCȚC%oPμPNȚCȚCȚ PI 1-PsPj CȚP»PsPIPμ PSP.

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	0032	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