

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.

Студент гр. 0383

Пенкин М.В.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Изучить работу программ hello1.asm и hello2.asm, выводящих на экран текст сообщения, с помощью отладчика. Представить протокол с фиксацией всех выполняемых действий и полученных результатов. Сравнить результаты прогона программ в отладчике и объяснить различия в программах.

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.

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.

Начальное содержимое сегментных регистров (CS) = 1A05, (DS) = 19F5, (ES) = 19F5 и (SS) = 1A0A.

Таблица 1. Протокол выполнения программы hello1

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 (DX) = 0000 (DS) = 19F5 (IP) = 0010	(AX) = 1A07 (DX) = 0000 (DS) = 19F5 (IP) = 0013
0013	mov ds, ax	8ED8	(AX) = 1A07 (DX) = 0000 (DS) = 19F5 (IP) = 0013	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (IP) = 0015
0015	mov dx, 0000	BA0000	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (IP) = 0015	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (IP) = 0018
0018	mov ah, 09	B409	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (IP) = 0018	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (IP) = 001A
001A	int 21	CD21	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (IP) = 001A	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (IP) = 001C
001C	mov ah, 4C	B44C	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (IP) = 001C	(AX) = 4C07 (DX) = 0000 (DS) = 1A07 (IP) = 001E
001E	int 21	CD21	(AX) = 4C07 (DX) = 0000 (DS) = 1A07 (IP) = 001E	Программа завершилась

Таблица 2. Протокол выполнения программы hello2

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0005	push ds	1E	(AX) = 0000 (DX) = 0000 (DS) = 19F5	(AX) = 0000 (DX) = 0000 (DS) = 19F5

			(SS) = 1A05 (CS) = 1A0A (SP) = 0018 (IP) = 0005 Stack +0 0000	(SS) = 1A05 (CS) = 1A0A (SP) = 0016 (IP) = 0006 Stack +0 19F5
0006	sub ax, ax	2BC0	(AX) = 0000 (DX) = 0000 (DS) = 19F5 (SS) = 1A05 (CS) = 1A0A (SP) = 0016 (IP) = 0006 Stack +0 19F5	(AX) = 0000 (DX) = 0000 (DS) = 19F5 (SS) = 1A05 (CS) = 1A0A (SP) = 0016 (IP) = 0008 Stack +0 19F5
0008	push ax	50	(AX) = 0000 (DX) = 0000 (DS) = 19F5 (SS) = 1A05 (CS) = 1A0A (SP) = 0016 (IP) = 0008 Stack +0 19F5	(AX) = 0000 (DX) = 0000 (DS) = 19F5 (SS) = 1A05 (CS) = 1A0A (SP) = 0014 (IP) = 0009 Stack +0 0000 Stack +2 19F5
0009	mov ax, 1A07	B8071A	(AX) = 0000 (DX) = 0000 (DS) = 19F5 (SS) = 1A05 (CS) = 1A0A (SP) = 0014 (IP) = 0009 Stack +0 0000 Stack +2 19F5	(AX) = 1A07 (DX) = 0000 (DS) = 19F5 (SS) = 1A05 (CS) = 1A0A (SP) = 0014 (IP) = 000C Stack +0 0000 Stack +2 19F5
000C	mov ds, ax	8ED8	(AX) = 1A07 (DX) = 0000 (DS) = 19F5 (SS) = 1A05 (CS) = 1A0A (SP) = 0014 (IP) = 000C Stack +0 0000 Stack +2 19F5	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (SS) = 1A05 (CS) = 1A0A (SP) = 0014 (IP) = 000E Stack +0 0000 Stack +2 19F5
000E	mov dx, 0000	BA0000	(AX) = 1A07	(AX) = 1A07

			(DX) = 0000 (DS) = 1A07 (SS) = 1A05 (CS) = 1A0A (SP) = 0014 (IP) = 000E Stack +0 0000 Stack +2 19F5	(DX) = 0000 (DS) = 1A07 (SS) = 1A05 (CS) = 1A0A (SP) = 0014 (IP) = 0011 Stack +0 0000 Stack +2 19F5
0011	call 0000	E8ECFF	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (SS) = 1A05 (CS) = 1A0A (SP) = 0014 (IP) = 0011 Stack +0 19F5 Stack +2 19F5	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (SS) = 1A05 (CS) = 1A0A (SP) = 0012 (IP) = 0000 Stack +0 0014 Stack +2 0000 Stack +4 19F5
0000	mov ah 09	B409	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (SS) = 1A05 (CS) = 1A0A (SP) = 0012 (IP) = 0000 Stack +0 0014 Stack +2 0000 Stack +4 19F5	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (SS) = 1A05 (CS) = 1A0A (SP) = 0012 (IP) = 0002 Stack +0 0014 Stack +2 0000 Stack +4 19F5
0002	int 21	CD21	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (SS) = 1A05 (CS) = 1A0A (SP) = 0012 (IP) = 0002 Stack +0 0014 Stack +2 0000 Stack +4 19F5	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (SS) = 1A05 (CS) = 1A0A (SP) = 0012 (IP) = 0004 Stack +0 0014 Stack +2 0000 Stack +4 19F5
0004	ret	C3	(AX) = 0907 (DX) = 0000 (DS) = 1A07	(AX) = 0907 (DX) = 0000 (DS) = 1A07

			(SS) = 1A05 (CS) = 1A0A (SP) = 0012 (IP) = 0004 Stack +0 0014 Stack +2 0000 Stack +4 19F5	(SS) = 1A05 (CS) = 1A0A (SP) = 0014 (IP) = 0014 Stack +0 0000 Stack +2 19F5
0014	mov dx, 0010	BA1000	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (SS) = 1A05 (CS) = 1A0A (SP) = 0014 (IP) = 0014 Stack +0 0000 Stack +2 19F5	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (SS) = 1A05 (CS) = 1A0A (SP) = 0014 (IP) = 0017 Stack +0 0000 Stack +2 19F5
0017	call 0000	E8E6FF	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (SS) = 1A05 (CS) = 1A0A (SP) = 0014 (IP) = 0017 Stack +0 0000 Stack +2 19F5	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (SS) = 1A05 (CS) = 1A0A (SP) = 0012 (IP) = 0000 Stack +0 001A Stack +2 0000 Stack +4 19F5
0000	mov ah, 09	B409	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (SS) = 1A05 (CS) = 1A0A (SP) = 0012 (IP) = 0000 Stack +0 001A Stack +2 0000 Stack +4 19F5	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (SS) = 1A05 (CS) = 1A0A (SP) = 0012 (IP) = 0002 Stack +0 001A Stack +2 0000 Stack +4 19F5
0002	int 21	CD21	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (SS) = 1A05 (CS) = 1A0A	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (SS) = 1A05 (CS) = 1A0A

			SP) = 0012 (IP) = 0002 Stack +0 001A Stack +2 0000 Stack +4 19F5	(SP) = 0012 (IP) = 0004 Stack +0 001A Stack +2 0000 Stack +4 19F5
0004	ret	C3	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (SS) = 1A05 (CS) = 1A0A (SP) = 0012 (IP) = 0004 Stack +0 001A Stack +2 0000 Stack +4 19F5	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (SS) = 1A05 (CS) = 1A0A (SP) = 0014 (IP) = 001A Stack +0 0000 Stack +2 19F5
001A	ret far	CB	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (SS) = 1A05 (CS) = 1A0A (SP) = 0014 (IP) = 001A Stack +0 0000 Stack +2 19F5	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (SS) = 1A05 (CS) = 19F5 (SP) = 0018 (IP) = 0000 Stack +0 0000
0000	int 20	CD20	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (SS) = 1A05 (CS) = 19F5 (SP) = 0018 (IP) = 0000 Stack +0 0000	Программа завершилась

Тексты исходных файлов программ см. в приложении А.

Тексты файлов диагностических сообщений см. в приложении Б.

Выводы.

В ходе лабораторной работы ознакомились с эмулятором DOSBox, научились работать с отладчиком AFD Pro и рассмотрели простейшие

программы на языке ассемблера. Был рассмотрен базовый синтаксис языка и изучены разные подходы к реализации задания, в данном случае — печати строки.

ПРИЛОЖЕНИЕ А

ТЕКСТЫ ИСХОДНЫХ ФАЙЛОВ ПРОГРАММ

Название файла: hello1.asm

```
; 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.0383 - Пенкин М.В.',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 4350 - $'
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

ТЕКСТЫ ФАЙЛОВ ДИАГНОСТИЧЕСКИХ СООБЩЕНИЙ

Название файла: hello1.lst

Microsoft (R) Macro Assembler Version 5.10
20:18:55

9/12/21

Page

1-1

```
; HELLO1.ASM - упрощенная версия учебной прог-
аммы лаб.раб. N1
;               по дисциплине "Архитектура комп-
ьютера"
; *****
*****
; Назначение: Программа формирует и выводит на
экран приветствие
;               пользователя с помощью функции ДО
С "Вывод строки"
;               (номер 09 прерывание 21h), котора
я:
;               - обеспечивает вывод на экран ст
роки символов,
;               заканчивающейся знаком "$";
;               - требует задания в регистре ah
номера функции=09h,
;               а в регистре dx - смещения а
дреса 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
```


END

Microsoft (R) Macro Assembler Version 5.10
20:18:55

9/12/21

Symbols-1

Segments and Groups:

Class	N a m e	Length	Align	Combine
DGROUP	GROUP		
_DATA	002C 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 + 461315 Bytes symbol space free

0 Warning Errors

0 Severe Errors

Название файла: hello2.lst

1-1

```

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

= 0024          EOFLine EQU '$'          ; Определение
СИМВОЛЬ        ной константы
; "Конец строки"

; Стек программы

ASSUME CS:CODE, SS:AStack

0000          AStack SEGMENT STACK
0000 000C[      DW 12 DUP(?)          ; Отводится 12
слов п        амяти
              ]
              ]

0018          AStack ENDS

; Данные программы

0000          DATA SEGMENT

; Директивы описания данных

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

                ; Код программы

0000             CODE            SEGMENT
                ; Процедура печати строки
0000             WriteMsg PROC NEAR
0000 B4 09                mov     AH,9
0002 CD 21                int     21h ; Вызов функции DOS по
пре
                рыванию
0004 C3                ret
0005             WriteMsg ENDP

                ; Головная процедура
0005             Main          PROC FAR
0005 1E                push    DS      ; \ Сохранение адреса
                начала PSP в стеке
0006 2B C0                sub     AX,AX      ; > для
последующего в
Microsoft (R) Macro Assembler Version 5.10          9/2/21
11:22:44

```

Page

1-2

```

                ;остановления по
0008 50                push    AX      ; / команде ret, завер
                шающей процедуру.
0009 B8 ---- R          mov     AX,DATA      ;
Загрузка
                сегментного
000C 8E D8                mov     DS,AX      ;
регистра
                данных.
000E BA 0000 R          mov     DX, OFFSET HELLO ; Вывод
на
                экран первой
0011 E8 0000 R          call    WriteMsg      ; строки
пр
                иветствия.
0014 BA 0010 R          mov     DX, OFFSET GREETING ; Вывод
на
                экран второй

```



```

0017  E8 0000 R          call WriteMsg          ; строки
пр
                                иветствия.
001A  CB                  ret                    ; Выход в D
                                OS по команде,
                                ; находящейся
                                ся в 1-ом слове PSP.
001B          Main      ENDP
001B          CODE      ENDS
                                END Main
Microsoft (R) Macro Assembler Version 5.10      9/2/21
11:22:44

```

Symbols-1

Segments and Groups:

Class	N a m e	Length	Align	Combine
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

47978 + 461297 Bytes symbol space free

0 Warning Errors

0 Severe Errors