

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №1
по дисциплине «ЭВМ»
Тема: Трансляции, отладка и выполнение программ на языке
Ассемблера.

Студентка гр. 1303

Хабибуллина А.М

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

Изучение основных принципов трансляции, отладки и выполнения 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`

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. Подключение в DOSBOX папки MASM, где хранятся исходные файлы программ.
2. Трансляция программы `>masm hello1.asm`. Создание объектного файла и листинга: `hello1.obj`, `lst1.lst`
3. Компонуем загрузочный модуль `>link hello1.obj`
4. Запускаем программу `>hello1.exe`
5. С помощью отладчика `afd` проходимся пошагово по программе.
6. `>afipro hello1.exe`

Таблицы с пошаговым рассмотрением программ представлены ниже: для `hello1.asm` и для `hello2.asm`.

hello1.asm

Адрес	Символический код	16-ричный	Содержимое регистров и

КОМАНДЫ	КОМАНДЫ	КОД команды	ячеек памяти	
			До выполнения	После выполнения
0010	MOV AX, 1A07	B8071A	(AX) = 0000 (DS) = 19F5 (IP) = 0010	(AX) = 1A07 (DS) = 19F5 (IP) = 0013
0013	MOV DS, AX	8ED8	(AX) = 1A07 (DS) = 19F5 (IP) = 0013	(AX) = 1A07 (DS) = 1A07 (IP) = 0015
0015	MOV DX, 0000	BA0000	(AX) = 1A07 (DS) = 1A07 (IP) = 0015 (CX) = 0000	(AX) = 1A07 (DS) = 1A07 (IP) = 0018 (CX) = 004B
0018	MOV AH, 09	B409	(AX) = 1A07 (DS) = 1A07 (IP) = 0018	(AX) = 0907 (DS) = 1A07 (IP) = 001A
001A	INT 21	B44C	(AX) = 0907 (DS) = 1A07 (IP) = 001A	(AX) = 0907 (DS) = 1A07 (IP) = 001C
001C	MOV AH, 4C	B44C	(AX) = 0907 (DS) = 1A07 (IP) = 001C	(AX) = 4C07 (DS) = 1A07 (IP) = 001E
001E	INT 21	CD21	(AX) = 4C07 (DS) = 1A07 (IP) = 001E (CX) = 004B	(AX) = 0000 (DS) = 19F5 (IP) = 0010 (CX) = 0000

hello2.asm

Адрес команды	Символический код команды	16- ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0005	PUSH DS	1E	(AX) = 0000 (DS) = 19F5 (IP) = 0005 (SP) = 0018 STACK+0=0000	(AX) = 0000 (DS) = 19F5 (IP) = 0006 (SP) = 0016 STACK+0=19F5
0006	SUB AX, AX	2BC0	(IP) = 0006	(IP) = 0008
0008	PUSH AX	50	(IP) = 0008 (SP) = 0016 STACK+0=A9F5 STACK+2=0000	(IP) = 0009 (SP) = 0014 STACK+0=0000 STACK+2=19F5
0009	MOV AX, 1A07	B8071A	(AX) = 0000 (DS) = 19F5 (IP) = 0009	(AX) = 1A07 (DS) = 19F5 (IP) = 000C
000C	MOV DS, AX	8ED8	(AX) = 1A07 (DS) = 19F5 (IP) = 000C	(AX) = 1A07 (DS) = 1A07 (IP) = 000E
000E	MOV DX, 0000	BA0000	(AX) = 1A07 (DS) = 1A07 (IP) = 000E	(AX) = 1A07 (DS) = 1A07 (IP) = 0011
0011	CALL 0000	E8ECFF	(IP) = 0011 (SP) = 0014 STACK+0=0000 STACK+2=19F5	(IP) = 0000 (SP) = 0012 STACK+0=0014 STACK+2=0000

			STACK+4=0000	STACK+4=19F5
0000	MOV AH, 09	B409	(AX) = 1A07 (DS) = 1A07 (IP) = 0000	(AX) = 0907 (DS) = 1A07 (IP) = 0002
0002	INT 21	CD21	(IP) = 0002	(IP) = 0004
0004	RET	C3	(AX) = 0907 (DS) = 1A07 (IP) = 0004	(AX) = 0907 (DS) = 1A07 (IP) = 0014
0014	MOV DX, 0010	BA1000	(DX) = 0000 (IP) = 0014	(DX) = 0010 (IP) = 0017
0017	CALL 0000	E8E6FF	(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	(IP) = 0004 (SP) = 0012 STACK+0=001A STACK+2=0000 STACK+4=19F5	(IP) = 001A (SP) = 0014 STACK+0=0000 STACK+2=19F5 STACK+4=0000
001A	RET FAR	CB	(IP) = 001A (SP) = 0014 STACK+2=19F5	(IP) = 0000 (SP) = 0018 STACK+2=0000
0000	INT 20	CD20	(AX) = 0907 (DX) = 0010	(AX) = 0000 (DX) = 0000

			(IP) = 0000 (CX) = 007B (DS) = 1A07	(IP) = 0005 (CX) = 0000 (DS) = 19F5
--	--	--	---	---

Выводы.

В ходе лабораторной работы было изучено, как работает программа на языке Ассемблер. Была произведена трансляция, линковка и выполнение программы.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

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.7303 - Иванов И.И.',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

```

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. Листинг успешной трансляции
программами.**

hello1.lst

hello2.lst