

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

Ильин Д.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Изучать как устроены трансляция, отладка 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.

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.

Ход работы

1. Часть 1

- 1.1. Просмотрена программа hello1.asm и разобрана структура и реализация каждого сегмента программы.
- 1.2. Изменена символьная строка Greeting в сегменте данных на „Привет мир“
- 1.3. Протранслирована программа с помощью MASM.EXE с созданием объектного файла hello1.obj и файлом листинга hello1.lst без ошибок.
- 1.4. Скомпонован загрузочный модуль с созданием исполняемого файла hello1.exe , также без ошибок.
- 1.5. Запущена программа в автоматическом режиме. Программа работает корректно , выводит строку „Вас приветствует ст.гр.0382 - Ильин Д.А.“.
- 1.6. Запущено выполнение программы под управлением отладчика с пошаговым выполнением и занесением данных в таблицу 2.

Начальное значение сегментных регистров:

CS = 1A05 ; DS = 19F5 ;

ES = 19F5 ; SS = 1A08 ;

Таблица 1 - Отладка 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, 000	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 = 14A1

2. Часть 2

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

Результат пошагового выполнения находится в таблице 3.

Начальное значение сегментных регистров:

CS = 1A0A ; DS = 19F5 ;

ES = 19F5 ; SS = 1A05 ;

Таблица 2 - Отладка 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	IP = 0005

			SP = 0018 Stack +0 0000 +2 0000 +4 0000 +6 0000	SP = 0016 Stack +0 19F5 +2 0000 +4 0000 +6 0000
0006	SUB AX, AX	2B C0	IP = 0005	IP = 0008
0008	PUSH AX	50	IP = 0008 PS = 0016 Stack +0 19F5 +2 0000 +4 0000 +6 0000	IP = 0009 PS = 0014 Stack +0 0000 +2 19F5 +4 0000 +6 0000
0009	MOV AX, 1A07	B8071A	IP = 0009 AX = 0000	IP = 000C 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	IP = 0014 SP = 0014 Stack +0 0000

			+2 0000 +4 19F5 +6 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 Stack +0 001A +2 0000 +4 19F5 +6 0000	IP = 001A SP = 0014 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

Выводы.

В ходе работы был изучен синтаксис языка Ассемблера и освоены трансляция, выполнение и отладка программ.

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 'Привет мир',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: 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.0382 - Ильин Д.А.',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 'Вас приветствует ст.гр.0382 - Ильин Д.А. $'
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

```