**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. 1303 |  | Самохин К.А. |
| Преподаватель |  | Ефремов М.А. |

Санкт-Петербург

2022

## Цель работы.

Изучить и применить на практике трансляцию, отладку и выполнение программ на языке Асемблера.

## Задание.

​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.

1. Разобраться в структуре и реализации каждого сегмента программы. Непонятные фрагменты прояснить у преподавателя. Строку-приветствие преобразовать в соответствии со своими личными данными.
2. Загрузить файл hello1.asm из каталога Задания в каталог Masm.
3. Протранслировать программу с помощью строки

> masm hello1.asm

c 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.

1. Скомпоновать загрузочный модуль с помощью строки

> link hello1.obj

с созданием карты памяти и исполняемого файла hello1.exe.

1. Выполнить программу в автоматическом режиме путем набора строки

> hello1.exe

убедиться в корректности ее работы и зафиксировать результат выполнения в протоколе.

1. 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.

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Адрес  Команды | Символический  код команды | 16-ричный  код команды | Содержимое регистров и ячеек памяти | |
| до выполнения . | После выполнения |
| 0003 | Mov DS, AX | 8E D8 | (AX) = 2D87  (DS) = 2D75  (IP) = 0003 | (AX) = 2D87  (DS) = 2D87  (IP) = 0005 |

## Выполнение работы.

Часть 1:

1. В текстовом редакторе была просмотрен код программы hello1.asm

2. Строка — приветствие была преобразована в соответствии с моими личными данными, каждый сегмент программы был разобран.

3. Файл hello1.asm был загружен в папку DOSBOX в эмуляторе MS-DOS

4. Программа была протранслированна с помощью команды:

> masm hello1.asm

5. Скомпонован загрузочный модуль с созданием карты памяти и исполняемого файла hello1.exe. с помощью строки:

> link hello1.obj

6. Программа выполнена в автоматическом режиме путем набора строки:

> hello1.exe

Программа выполнена корректно

7. Программа была запущена под управлением отладчика AFDPRO:

> afdpro hello1.exe

Начальные значения регистров:

CS = 1A05

DS = 19F5

ES = 19F5

SS = 1A0A

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Адрес  Команды | Символический  код команды | 16-ричный  код команды | Содержимое регистров и ячеек памяти | |
| до выполнения | После выполнения |
| 0010 | MOV AX, 1A07 | B8AE11 | (AX) = 0000  (IP) = 0010 | (AX) = 1A07  (IP) = 0013 |
| 0013 | MOV DS, AX | 8ED8 | (DS) = 19F5  (IP) = 0013  (AX) = 1A07 | (DS) = 1A07  (IP) = 0015  (AX) = 1A07 |
| 0015 | MOV DX, 0000 | BA0000 | (IP) = 0015 | (IP) = 0018 |
| 0018 | MOV AH, 9 | B409 | (AX) = 1A07  (AH) = 1A  (IP) = 0018 | (AX) = 0907  (AH) = 09  (IP) = 001A |
| 001A | INT 21H | CD21 | (IP) = 001A | (IP) = 001C |
| 001C | MOV AH, 4CH | B44C | (AH) = 09  (AX) = 0907  (IP) = 001C | (AH) = 4C (AX) = 4C07  (IP) = 001E |
| 001E | INT 21H | CD21 | (AX) = 4C07  (DS) = 1A07  (IP) = 001E | (AX) = 0000  (DS) = 19F5  (IP) = 0010 |

## Часть 2.

1. В текстовом редакторе была просмотрен код программы hello2.asm2

2. Строка — приветствие была преобразована в соответствии с моими личными данными, каждый сегмент программы был разобран.

3. Файл hello2.asm был загружен в папку DOSBOX в эмуляторе MS-DOS

4. Программа была протранслированна с помощью команды:

> masm hello2.asm

5. Скомпонован загрузочный модуль с созданием карты памяти и исполняемого файла hello2.exe. с помощью строки:

> link hello2.obj

6. Программа выполнена в автоматическом режиме путем набора строки:

> hello2.exe

Программа выполнена корректно

7. Программа была запущена под управлением отладчика AFDPRO:

> afdpro hello2.exe

Начальные значения регистров:

CS = 1A0A

DS = 19F5

ES = 19F5

SS = 1A05

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Адрес  Команды | Символический  код команды | 16-ричный  код команды | Содержимое регистров и ячеек памяти | |
| до выполнения | После выполнения |
| 0005 | PUSH DS | 1E | (SP) = 0018  (IP) = 0005  Stack +0 = 0000 | (SP) = 0016  (IP) = 0006  Stack +0 = 19F5 |
| 0006 | SUB AX,AX | 2BC0 | (IP) = 0006  (AX) = 0000  (ZF) = 0  (PF) = 0 | (IP) = 0008  (AX) = 0000  (ZF) = 1  (PF) = 1 |
| 0008 | PUSH AX | 50 | (SP) = 0016  (IP) = 0008  Stack +0 = 19F5  Stack +2 = 0000 | (SP) = 0014  (IP) = 0009  Stack +0 = 0000  Stack +2 = 19F5 |
| 0009 | MOV AX,DATA | B8AE11 | (AX) = 0000  (IP) = 0009 | (AX) = 1A07  (IP) = 000C |
| 000C | MOV DS,AX | 8ED8 | (DS) = 19F5  (IP) = 000C | (DS) = 1A07  (IP) = 000E |
| 000E | MOV DX, 0000 | BA0000 | (DX) = 0000  (IP) = 000E | (DX) = 0000  (IP) = 0011 |
| 0011 | CALL 0000 | E8ECFF | (SP) = 0014  (IP) = 0011  Stack +0 = 0000  Stack +2 = 19F5  Stack +4 = 0000 | (SP) = 0012  (IP) = 0000  Stack +0 = 0014  Stack +2 = 0000  Stack +4 = 19F5 |
| 0000 | MOV AH,9 | B409 | (AH) = 1A  (AX) = 1A07  (IP) = 0000 | (AH) = 09  (AX) = 0907  (IP) = 0002 |
| 0002 | INT 21H | CD21 | (IP) = 0002 | (IP) = 0004 |
| 0004 | RET | C3 | (SP) = 0012  (IP) = 0004  Stack +0 = 0014  Stack +2 = 0000  Stack +4 = 19F5 | (SP) = 0014  (IP) = 0014  Stack +0 = 0000  Stack +2 = 19F5  Stack +4 = 0000 |
| 0014 | MOV DX, 0010 | BA0D00 | (DX) = 0000  (IP) = 0014 | (DX) = 0010  (IP) = 0017 |
| 0017 | CALL 0000 | E8E6FF | (SP) = 0014  (IP) = 0017  Stack +0 = 0000  Stack +2 = 19F5  Stack +4 = 0000 | (SP) = 0012  (IP) = 0000  Stack +0 = 001A  Stack +2 = 0000  Stack +4 = 19F5 |
| 0000 | MOV AH,9 | B409 | (AH) = 09  (IP) = 0000 | (AH) = 09  (IP) = 0002 |
| 0002 | INT 21H | CD21 | (IP) = 0002 | (IP) = 0004 |
| 0004 | RET | C3 | (SP) = 0012  (IP) = 0004  Stack +0 = 001A  Stack +2 = 0000  Stack +4 = 19F5 | (SP) = 0014  (IP) = 001A  Stack +0 = 0000  Stack +2 = 19F5  Stack +4 = 0000 |
| 001A | RET | CB | (SP) = 0014  (CS) = 1A0A  (IP) = 001A  Stack +0 = 0000  Stack +2 = 19F5 | (SP) = 0018  (CS) = 19F5  (IP) = 0000  Stack +0 = 0000  Stack +2 = 0000 |
| 0000 | INT 20 | CD20 | (IP) = 0000 | (IP) = 0005 |

## Выводы.

В ходе выполнения работы нам удалось ознакомиться с трансляцией, линковкой и выполнением двух программ на языке Ассемблер. Также обе программы были успешно запущены и протестированы.

# Приложение А

; 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.1303 – Самохин К.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

**ПРИЛОЖЕНИЕ В**

; 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 1303 - $'

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