**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: Наименование темы**

|  |  |  |
| --- | --- | --- |
| Студент гр. 7383 |  | Медведев И. С. |
| Преподаватель |  | Кирьянчиков В. А. |

Санкт-Петербург

2017

Цель работы:

Освоить приемы работы с DOSBOX, изучить основные команды для компиляции, линковки и запуска программы, написанной на языке программирования Assembler.

Ход Работы:

С помощью FAR был открыт код программы в режиме редактирования и произведена замена строки, выводимой на экран на следующий текст «Вас приветствует ст. гр. 7383 – Медведев И. С.».

Программа протранслирована с помощью строки > masm HELLO1.asm с созданием объектного файла HELLO1.obj и файла диагностических сообщений (файла листинга) 1.lst. Синтаксических ошибок обнаружено не было.

Скомпонован загрузочный модуль (HELLO1.exe) с помощью строки >link HELLO1.obj с созданием загрузочного модуля (HELLO1.exe) и файла карты памяти (HELLO1.map). По карте памяти оценена длина и размещение сегментов памяти. Содержимое файла HELLO1.map представлено в таблице 1.

Таблица 1 - Содержимое файла HELLO1.map.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Start** | **Stop** | **Length** | **Name** | **Class** |
| 00000H | 0001FH | 00020H | \_TEXT | CODE |
| 00020H | 0004BH | 0002CH | \_DATA | DATA |
| 00050H | 0014FH | 00100H | STACK | STACK |

Выполнена программа в автоматическом режиме с помощью набора строки > HELLO1.exe. При этом на экран была выведена строка «Вас приветствует ст. гр. 7383 – Медведев И.С.»

Также программа была выполнена в пошаговом режиме под управлением отладчика: >afd HELLO1.exe с фиксацией содержимого используемых регистров и ячеек памяти до и после выполнения команды.

Обычные команды выполнены по F1, а прерывания с помощью клавиши F2, чтобы не входить внутрь обработчика прерываний.

Выход из отладчика - по команде quit.

Записано содержимое всех регистров процессора, включая сегментные, перед выполнением первой команды. Результаты прогона программы под управлением отладчика представлены в таблице 1.

Содержимое всех регистров процессора до выполнения первой команды:

AX 0000 SI 0000 CS 1594 IP 0010

BX 0000 DI 0000 DS 1584

CX 004Е BP 0000 ES 1584 HS 1584

DX 0000 SP 0100 SS 1599 FS 1584

Результат пошагового прохождения программы представлен в Приложении А.

Просмотрена программа HELLO2.asm в режиме редактирования. Строка приветствия была записана в виде: «Здравствуйте! Вас приветствует ст.гр. 7383 – Медведев И. С.»

С помощью транслятора masm протранслирована программа HELLO2.asm c созданием объектного файла - HELLO2.obj и файла диагностических сообщений (файла листинга) - 2.lst.

Скомпонован загрузочный модуль HELLO2.exe с помощью компоновщика link.

Содержимое файла HELLO2.map:

Start Stop Length Name Class

00000H 00017H 00018H ASTACK

00020H 0005BH 0003CH DATA

00060H 0007AH 0001BH CODE

Программа выполнена в автоматическом режиме.

Выполнена программа HELLO2 в пошаговом режиме под управлением отладчика: >afd HELLO2.exe с фиксацией содержимого используемых регистров и ячеек памяти до и после выполнения команды.

Обычные команды выполнены по F1, а прерывания Int- по F2, чтобы не входить внутрь обработчика прерываний.

Выход из отладчика- по команде Quit.

Записано содержимое всех регистров процессора, включая сегментные, перед выполнением первой команды. Результаты прогона программы под управлением отладчика представлены в таблице 2.

Содержимое всех регистров процессора до выполнения первой команды:

AX 0000 SI 0000 CS 159A IP 0005

BX 0000 DI 0000 DS 1584

CX 007B BP 0000 ES 1584 HS 1584

DX 0000 SP 0018 SS 1594 FS 1584

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **F Адрес команды** | **Символический код команды** | **16-ричный код команды** | **Содержимое регистров и ячеек памяти** | |
| **До выполнения команды** | **После выполнения команды** |
| 0005 | PUSH DS | 1E | IP=0005  SP=0018  Stack+0 0000  +2 0000  +4 0000  +6 0000 | IP=0006  SP=0016  Stack +0 1584  +2 0000  +4 0000  +6 0000 |
| 0006 | SUB AX,AX | 2BCO | AX=0000  IP=0006 | AX=0000  IP=0008 |
| 0008 | PUSH AX | 50 | SP=0016  IP=0008  Stack+0 1584  +2 0000  +4 0000  +6 0000 | SP=0014  IP=0009  Stack +0 0000  +2 1584  +4 0000  +6 0000 |
| 0009 | mov AX,1596 | B8AE11 | AX=0000  IP=0009 | AX=1596  IP=000C |
| 000C | mov DS,AX | 8ED8 | DS=1584  IP=000C | DS=1596  IP=000E |
| 000E | mov DX,0000 | BA0000 | DX=0000  IP=0015 | DX=0000  IP=0011 |
| 0011 | CALL 0000 | E8ECFF | SP=0014  IP=0011  Stack+0 0000  +2 1584  +4 0000  +6 0000 | SP=0012  IP=0000  Stack +0 0014  +2 0000  +4 1584  +6 0000 |
| 0000 | mov AH,09 | B409 | AX=1596  IP=0000 | AX=0996  IP=0002 |
| 0002 | int 21 | CD21 | AX=0996  IP=0002 | AX=0924  IP=0004 |
| 0004 | RET | C3 | SP=0012  IP=0004  Stack+0 0014  +2 0000  +4 1584  +6 0000 | SP=0014  IP=0014  Stack +0 0000  +2 1584  +4 0000  +6 0000 |
| 0014 | mov DX,0010 | BA1000 | DX=0000  IP=0014 | DX=0010  IP=0017 |
| 0017 | CALL 0000 | E8E6FF | SP=0014  IP=0017  Stack +0 0000  +2 1584  +4 0000  +6 0000 | SP=0012  IP=0000  Stack +0 001A  +2 0000  +4 1584  +6 0000 |
| 0000 | mov AH,09 | B409 | AX=0924  IP=0000 | AX=0924  IP=0002 |
| 0002 | int 21 | CD211 | IP=0002 | IP=0004 |
| 0004 | RET | C3 | IP=0004  Stack +0 001A  +2 0000  +4 1584  +6 0000 | IP=001A  Stack +0 0000  +2 1584  +4 0000  +6 0000 |
| 001A | RET FAR | CB | SP=0012  CS=159A  IP=001A  Stack +0 0000  +2 1584  +4 0000  +6 0000 | SP=0018  CS=1584  IP=0000  Stack +0 0000  +2 0000  +4 0000  +6 0000 |
| 0000 | INT 20 | CD20 | AX=0924  CX=007B  CS=1584  IP=0000 | AX=0000  CX=0000  CS=159A  IP=0005  Program terminated normally |

Таблица 2

**Вывод:** в ходе данной лабораторной работы мною была освоена трансляция, выполнение и отладка программ на языке Ассемблера процессора Intel x86. Результатом выполнения лабораторной работы являются программы HELLO1.exe и HELLO2.exe.

Содержимое 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.4381 – Лапцевич 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

Содержимое 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

AStack SEGMENT STACK

DW 12 DUP(?) ; Отводится 12 слов памяти

AStack ENDS

; Данные программы

DATA SEGMENT

; Директивы описания данных

HELLO DB 'Здравствуйте!', 0AH, 0DH,EOFLine

GREETING DB 'Вас приветствует ст.гр.4381 – Лапцевич Д.А.$'

DATA ENDS

; Код программы

CODE SEGMENT

ASSUME CS:CODE, DS:DATA, SS:AStack

; 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