**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра MO ЭВМ**

отчет

**по лабораторной работе №1**

**по дисциплине «Организация ЭВМ и систем»**

Тема: Трансляции, отладка и выполнение программ на языке Ассемблера **процессора Intel X86**.

|  |  |  |
| --- | --- | --- |
| Студент гр. 9383 |  | Рыбников Р.А. |
| Преподаватель |  | Ефремов М.А. |

Санкт-Петербург

2020

**Цель работы.**

Освоение трансляции, отладки и выполнения программ на языке Ассемблера процессора Intel X86.

**Задание.**

Часть 1.

1. Загрузить файл hello1.asm из каталога \лаборат\_работы в каталог \MASM.

2. Просмотреть программу в режиме редактирования, разобраться в структуре и реализации каждого сегмента программы. Непонятные фрагменты прояснить у преподавателя. Строку-приветствие преобразовать в соответствии со своими личными данными.

3. Протранслировать программу с помощью строки

> masm имя\_файла.asm

c созданием объектного файла имя\_файла.obj и файла диагностических сообщений (файла листинга) имя\_файла.lst. Объяснить и исправить синтаксические ошибки, если они будут обнаружены транслятором и представлены в файле листинга. Повторить трансляцию программы до получения объектного файла(модуля).

4. Скомпоновать загрузочный модуль (имя\_файла.exe) с помощью строки

> link имя\_файла.obj

c созданием загрузочного модуля (имя\_файла.exe) и файла карты памяти (имя\_файла.map). По карте памяти оценить размещение и длину сегментов программы.

5. Выполнить программу в автоматическом режиме путем набора строки

> имя\_файла.exe

и убедиться в ее работоспособности.

6. Выполнить программу hello1 в пошаговом режиме под управлением

отладчика: > afd имя\_файла.exe

с фиксацией содержимого используемых регистров и ячеек памяти до и после выполнения команды.

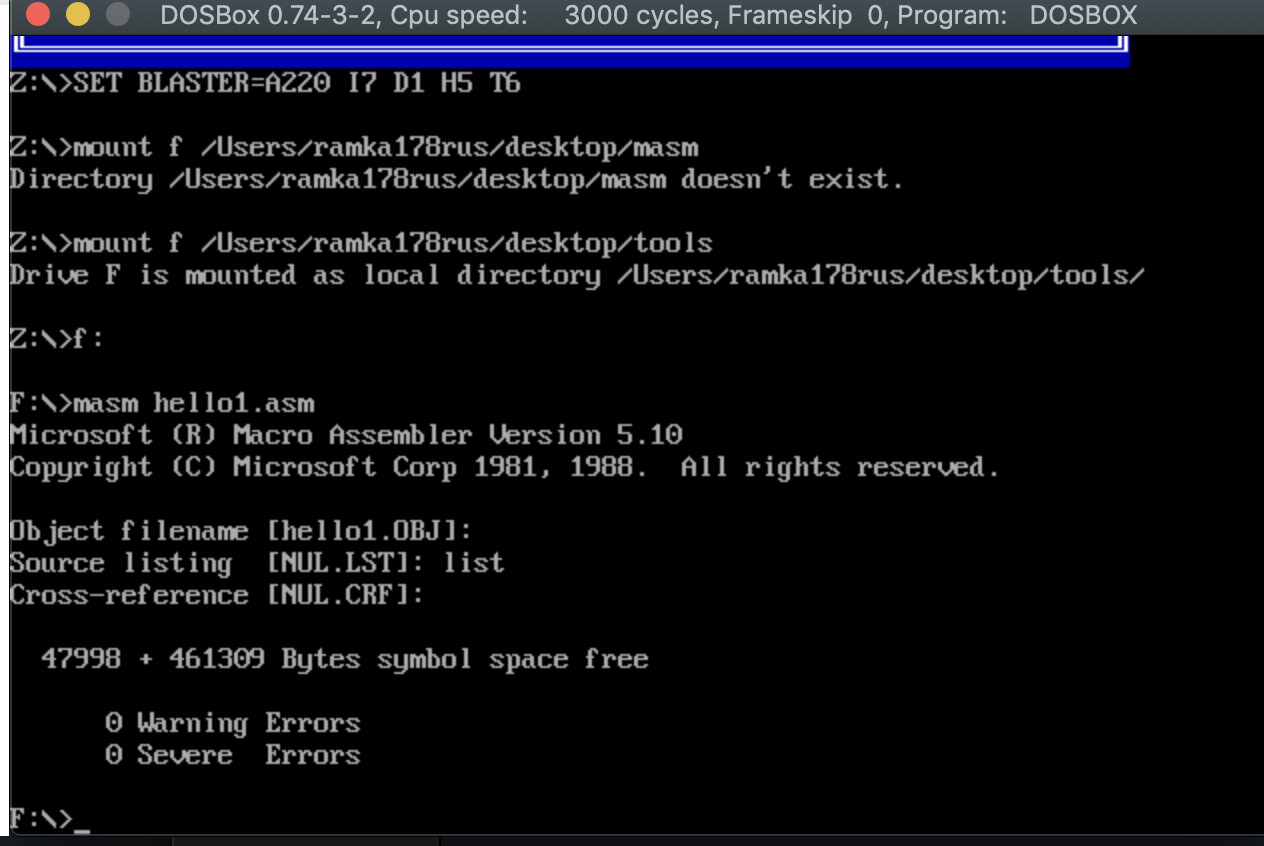
Результаты прогона программы под управлением отладчика должны быть представлены в виде, показанном на примере одной команды в табл.1, и подписаны преподавателем.

Часть 2.

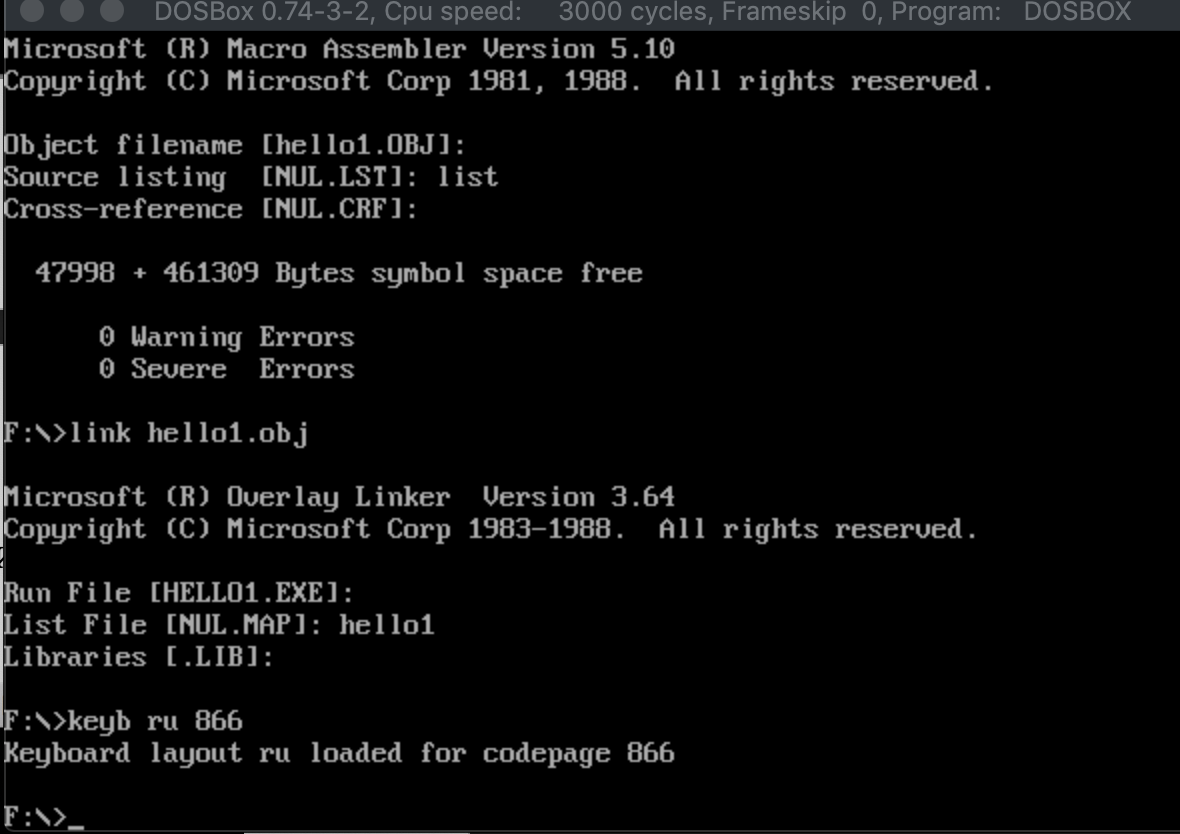
Выполнить пункты 1 - 7 части 1 настоящего задания применительно к программе hello2.asm,приведенной в каталоге Задания, которая выводит на экран приветствие пользователя с помощью процедуры WriteMsg, а также использует полное определение сегментов. Сравнить результаты прогона под управлением отладчика программ hello1 и hello2 и объяснить различия в размещении сегментов.

**Ход работы.**

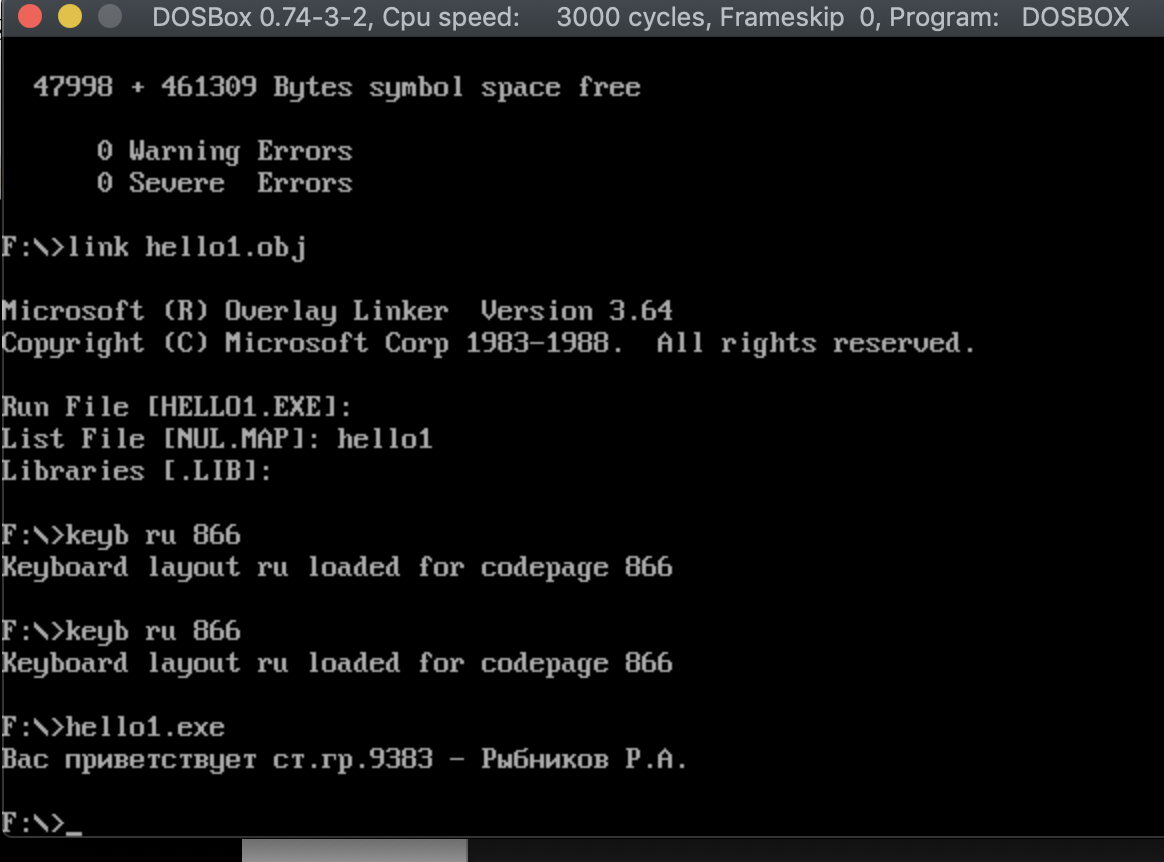
1. Загрузка файла HELLO1.ASM в каталог MASM.
2. Просмотр программы в режиме редактирования. В строку с приветствием были внесены изменения личных данных.
3. Трансляция программы с созданием объектного файла hello1.obj и файла диагностических сообщений list.lst с помощью строки masm hello1.asm.

****

1. Линковка загрузочного модуля hello1.exe с созданием загрузочного модуля hello1.exe и файла карты памяти hello1.map.

****

1. Выполнение программы в автоматическом режиме путем набора строки hello1.exe.

****

1. Выполнение программы в пошаговом режиме под управлением отладчика afd. Результаты представлены в таблице 1.

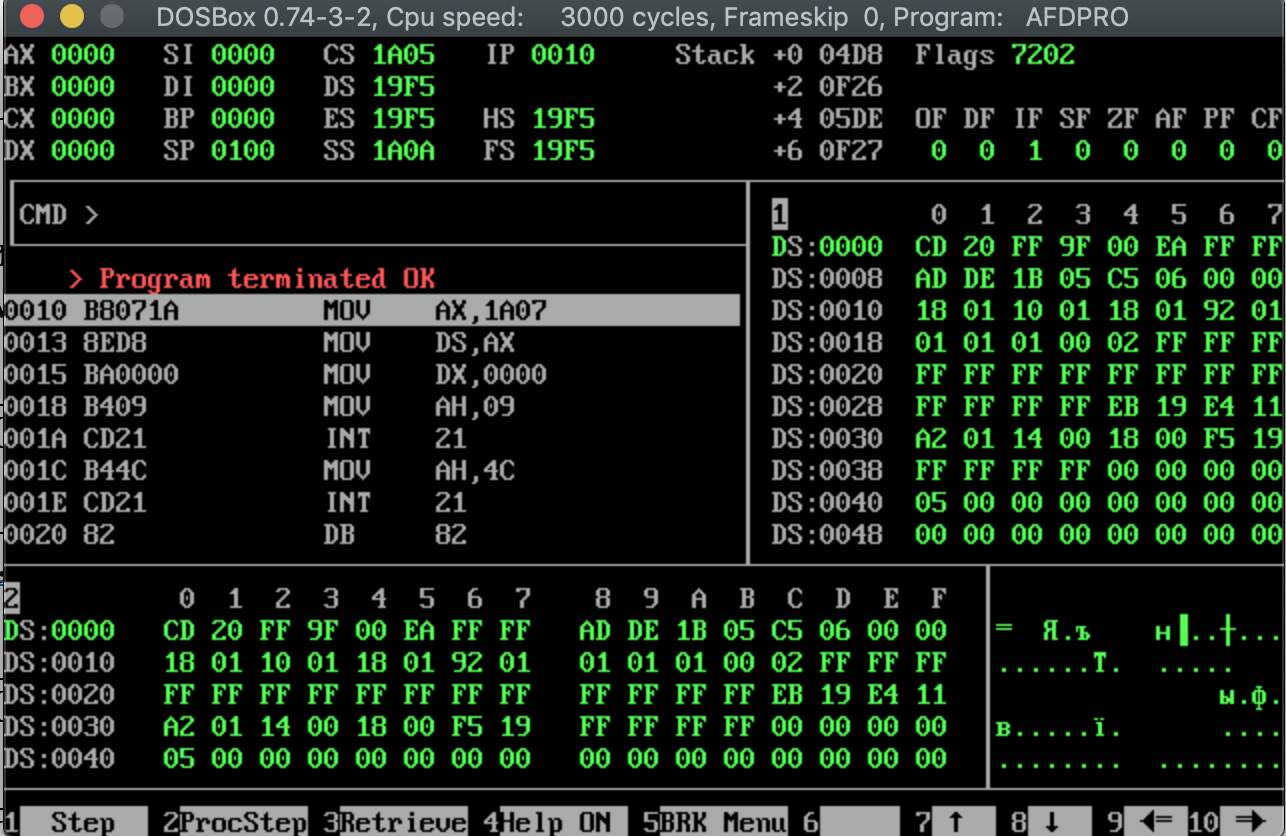
**Изображение выглядит как монитор, зеленый, сторона, стол

Автоматически созданное описание**

|  |  |
| --- | --- |
| Сегментный регистр | Начальное значение |
| CS | 1A05 |
| DS | 19F5 |
| ES | 19F5 |
| SS | 1A0A |
| HS | 19F5 |
| FS | 19F5 |

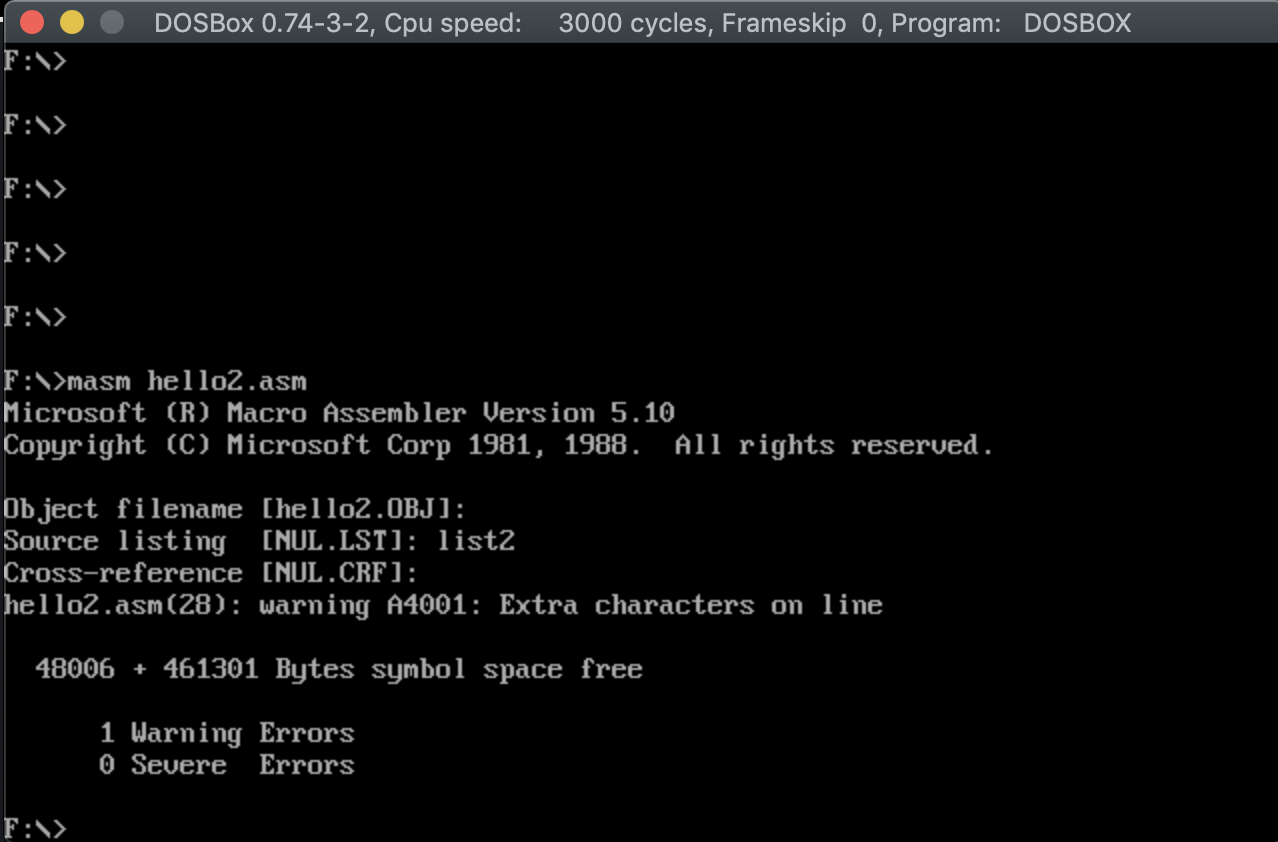
|  |  |  |
| --- | --- | --- |
| Stack | +0 | 04D8 |
| +2 | 0F26 |
| +4 | 05DE |
| +6 | 0F27 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Адрес команды | Символический код команды | 16-ричный команды | Содержимое регистров и ячеек памяти | |
| до выполнения  команды | после выполнения команды |
| 0010 | MOV ax,1A07 | B8071A | ax = 0000  ip = 0010 | Ax = 1A07  ip = 0013 |
| 0013 | MOV ds,ax | 8ed8 | ax = 1A07  ds = 19F5  ip = 0013 | ax = 1A07  ds = 1A07  ip = 0015 |
| 0015 | MOV dx,0000 | 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 = 001С | ax = 4C07  ip = 001E |
| 001E | INT 21 | cd21 | ip = 001E | 0010 |

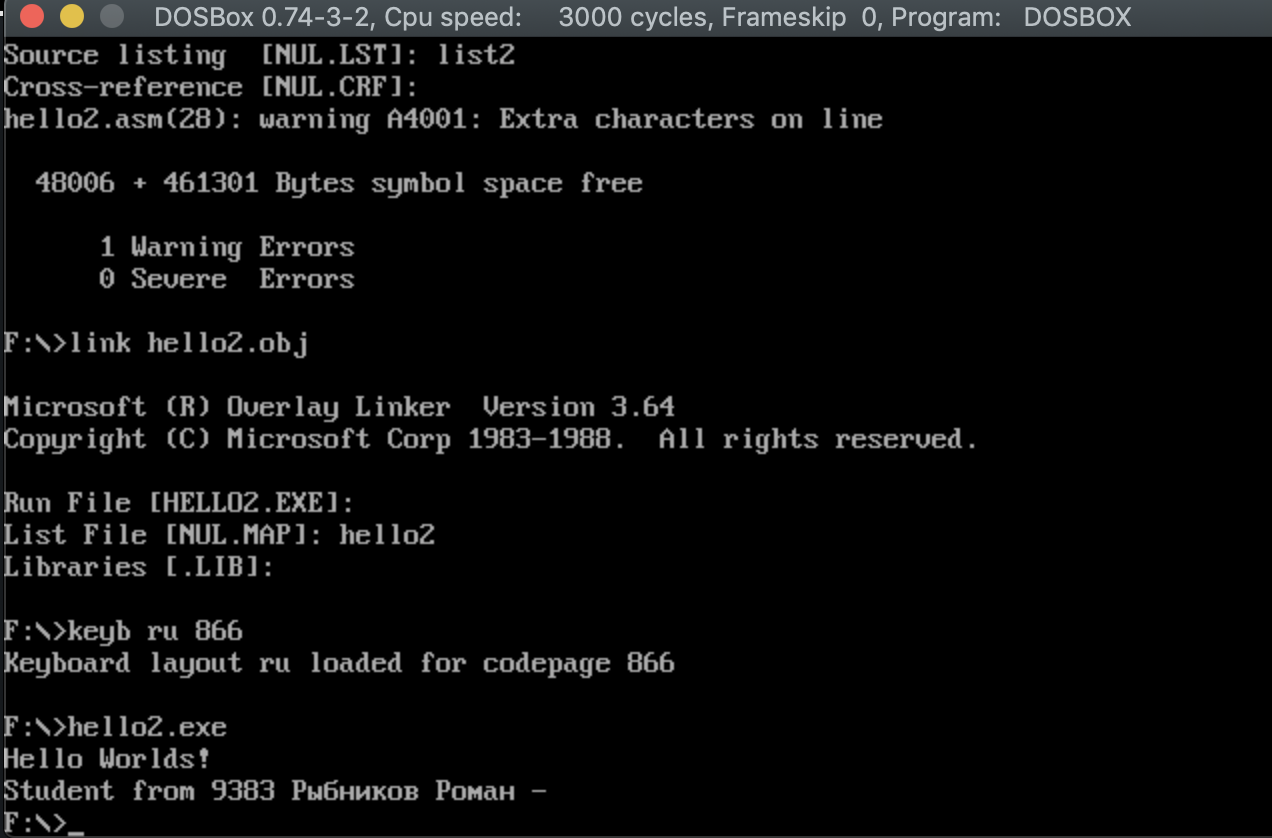
****

Часть 2.

1. Файл hello2.asm загружен в папку MASM.
2. Строка в файле отредактирована под личные данные.
3. Трансляция программы с созданием объектного файла hello2.obj и файла диагностических сообщений list.lst с помощью строки masm hello2.asm.



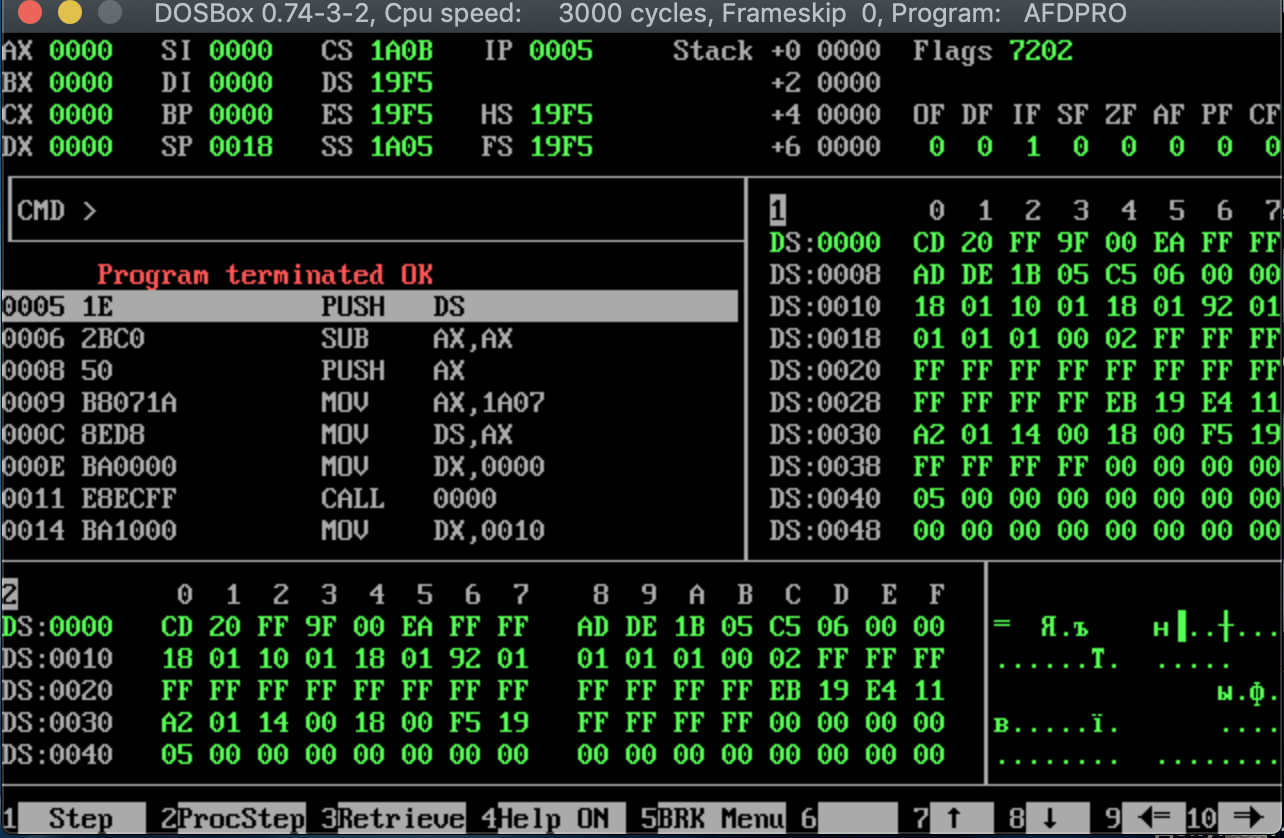
1. Линковка загрузочного модуля hello2.exe с созданием загрузочного модуля hello1.exe и файла карты памяти hello2.map.
2. Выполнение программы в автоматическом режиме путем набора строки hello1.exe.



1. Выполнение программы в пошаговом режиме под управлением отладчика afd. Результаты представлены в таблице 2.

|  |  |
| --- | --- |
| Сегментный регистр | Начальное значение |
| CS | 1A0B |
| DS | 19F5 |
| ES | 19F5 |
| SS | 1A05 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Адрес команды | Символический код команды | 16-ричный команды | Содержимое регистров и ячеек памяти | |
| до выполнения  команды | после выполнения команды |
| 0005 | PUSH DS | 1е | ip = 0005  Sp = 0018  stack:  +0 0000  +2 0000 +4 0000 +6 0000 | ip = 0006  Sp = 0016  stack:  +0 19F5  +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 19F5  +2 0000 +4 0000 +6 0000 | Sp = 0014  Ip = 0009  stack:  +0 0000  +2 119c +4 0000 +6 0000 |
| 0009 | MOV ax,1A07 | b8071A | 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 | 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 | ax = 1A07  ip = 0000 | ax = 0907  ip = 0002 |
| 0002 | INT 21 | cd21 | Ip = 0002 | ip = 0004 |
| 0004 | RET | c3 | ip = 0004  SP = 0012  stack:  +0 0014  +2 0000 +4 19F5 +6 0000 | ip = 0014  SP = 0014  stack:  +0 0000  +2 19F5 +4 0000 +6 0000 |
| 0014 | MOV dx,0010 | ba1000 | DX = 0000  ip = 0014 | DX = 0010  ip = 0017 |
| 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 | ax = 0907  ip = 0000 | ax = 0907  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 = 1A0B  stack:  +0 0000  +2 019F5  +4 000 +6 0000 | Ip = 0000  Sp = 0018  CS = 19F5  stack:  +0 0000  +2 0000  +4 0000 +6 0000 |
| 0000 | INt 20 | cd20 |  |  |



Результаты тестирования программ в отладчике afd имеют различия. В программе HELLO2 шагов в программе больше, к тому же тут используется процедура WriteMsg. Сегменты DOS размещены по-разному, так как в программе HELLO1 используется директива DOSSEG. Она разрешает переопределение сегментов DOS во время компоновки.

Вывод.

После выполнения работы были освоены принципы трансляции, выполнения и отладки программ на языке программирования Ассемблер процессора Intel x86.

ПРИЛОЖЕНИЕ.

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.9383 - Рыбников Р.A.',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

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 9383 Рыбников Роман - $'

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.

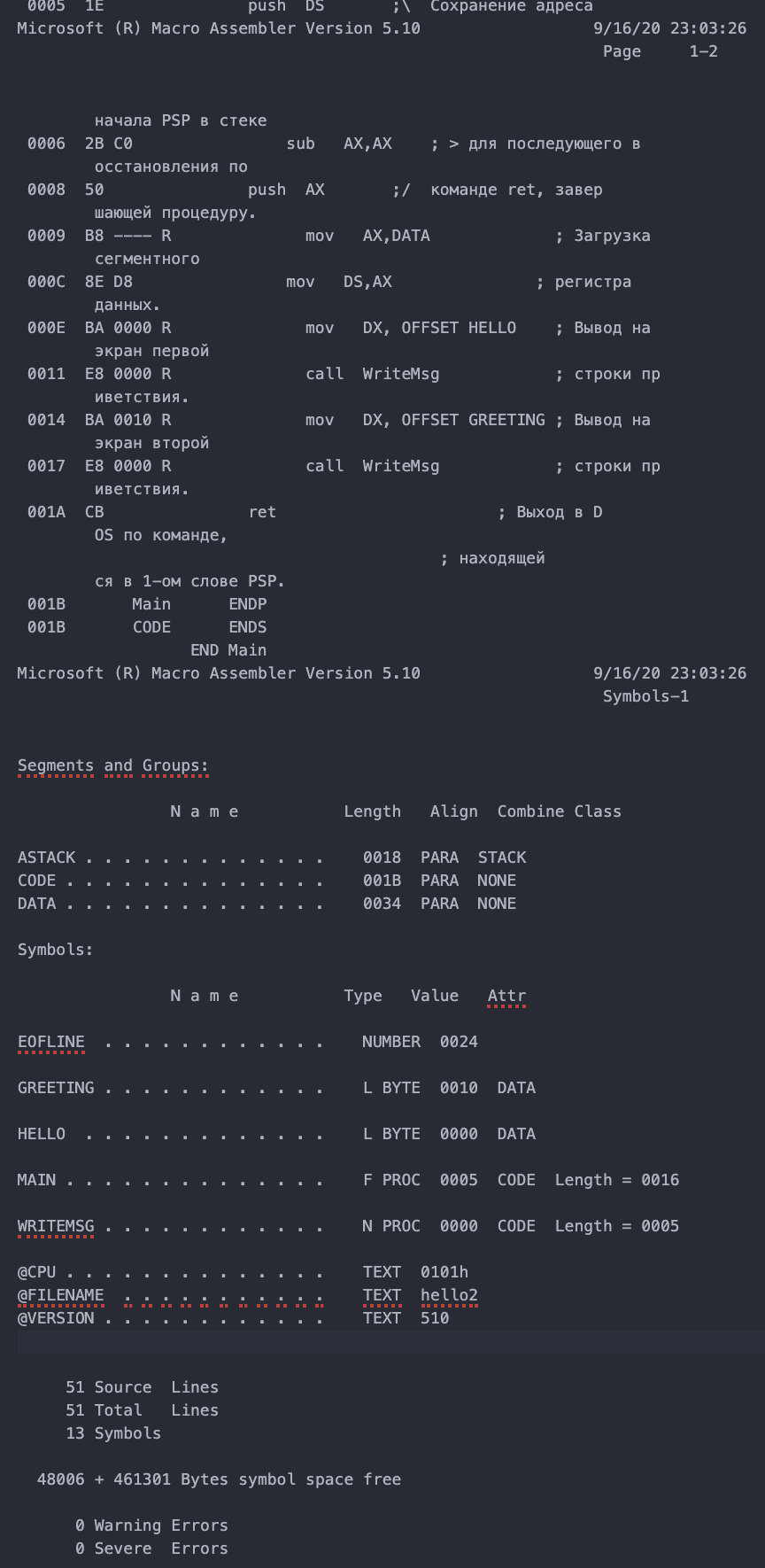
Main ENDP

CODE ENDS

END Main

list.lst





list2.lst



