МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»

**ОТЧЕТ ПО ПРАКТИКЕ ДИСЦИПЛИНЫ**

**МАШИННО ЗАВИСИМЫЕ ЯЗЫКИ ПРОГРАММИРОВАНИЯ**

Лабораторная № 1

студента 2 курса 241 группы

Синёва Вячеслава Андреевича

Преподаватель: Черноусова Елена Михайловна

Саратов 2023

Задание 1.1. Измените программы из примеров 1, 2 и 3 так, чтобы они выводили на экран ваши фамилию, имя и номер группы. Используя командные файлы (с расширением bat), подготовьте к выполнению и запустите 3 программы. Убедитесь, что они выводят на экран нужный текст и успешно завершаются.

Программа 1 (CLASSIC) z1.asm

stak segment stack 'stack' ;Начало сегмента стека

db 256 dup (?) ;Резервируем 256 байт для стека

stak ends ;Конец сегмента стека

data segment 'data' ;Начало сегмента данных

Inform db 'Sinew Vyacheslav 241$' ;Строка для вывода

data ends ;Конец сегмента данных

code segment 'code' ;Начало сегмента кода

assume CS:code,DS:data,SS:stak ;Сегментный регистр CS будет указывать на сегмент команд,

;регистр DS - на сегмент данных, SS – на стек

start: ;Точка входа в программу start

;Обязательная инициализация регистра DS в начале программы

mov AX,data ;Адрес сегмента данных сначала загрузим в AX,

mov DS,AX ;а затем перенесем из AX в DS

mov AH,09h ;Функция DOS 9h вывода на экран

mov DX,offset Inform ;Адрес начала строки 'Hello, World!' записывается в регистр DX

int 21h ;Вызов функции DOS

mov AX,4C00h ;Функция 4Ch завершения программы с кодом возврата 0

int 21h ;Вызов функции DOS

code ends ;Конец сегмента кода

end start ;Конец текста программы с точкой входа

Программа 2 (SMALL) z2.asm

.model small ;Модель памяти SMALL использует сегменты

;размером не более 64Кб

.stack 100h ;Сегмент стека размером 100h (256 байт)

.data ;Начало сегмента данных

Inform db 'Sinew Vyacheslav 241$'

.code ;Начало сегмента кода

start: ;Точка входа в программу start

;Предопределенная метка @data обозначает

;адрес сегмента данных в момент запуска программы,

mov AX, @data ;который сначала загрузим в AX,

mov DS,AX ;а затем перенесем из AX в DS

mov AH,09h

mov DX,offset Inform

int 21h

mov AX,4C00h

int 21h

end start

Программа 3 (TINY) z3.asm

.model tiny ;Модель памяти TINY, в которой код, данные и стек

;размещаются в одном и том же сегменте размером до 64Кб

.code ;Начало сегмента кода

org 100h ;Устанавливает значение программного счетчика в 100h

;Начало необходимое для COM-программы,

;которая загружается в память с адреса PSP:100h

start:

mov AH,09h

mov DX,offset Inform

int 21h

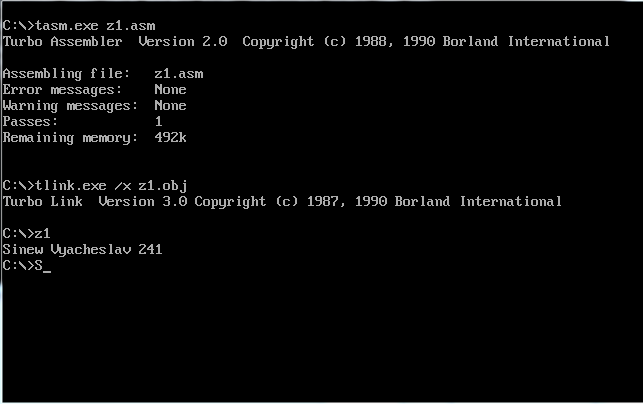
mov AX,4C00h

int 21h

;===== Data =====

Inform db 'Sinew Vyacheslav 241$'

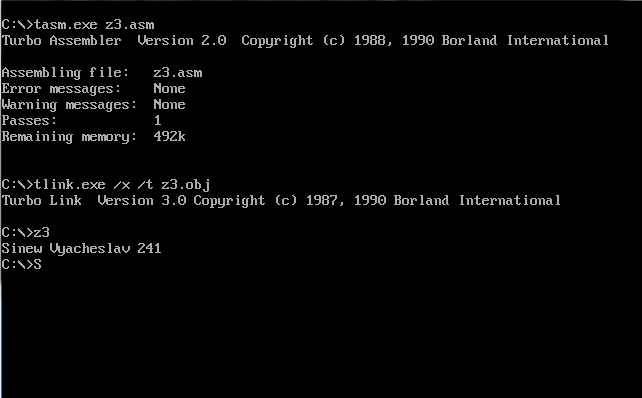
end start

Запуск программы z1

Запуск программы z2



Запуск программы z3

Командный файл make\_com.bat

cls

tasm.exe %1.asm

tlink.exe /x /t %1.obj

%1

Командный файл make\_exe.bat

cls

tasm.exe %1.asm

tlink.exe /x %1.obj

%1

Задание 1.2. Заполните таблицы трассировки для 3-х программ.

Таблица трассировки программы z1

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Шаг | Машинный код | Команда | Регистры | | | | | | | | |  |
| AX | BX | CX | DX | SP | DS | SS | CS | IP | CZSOPAID |
| 1 | B8BD48 | MOV AX, 48BD | 0000 | 0000 | 0000 | 0000 | 0100 | 489D | 48AD | 48BF | 0000 | 00000010 |
| 2 | 8ED8 | MOV DS, AX | 48BD | 0000 | 0000 | 0000 | 0100 | 489D | 48AD | 48BF | 0003 | 00000010 |
| 3 | B409 | MOV AH, 09 | 48BD | 0000 | 0000 | 0000 | 0100 | 48BD | 48AD | 48BF | 0005 | 00000010 |
| 4 | BA0000 | MOV DX, 0000 | 09BD | 0000 | 0000 | 0000 | 0100 | 48BD | 48AD | 48BF | 0007 | 00000010 |
| 5 | CD21 | INT 21 | 09BD | 0000 | 0000 | 0000 | 0100 | 48BD | 48AD | 48BF | 000A | 00000010 |
| 6 | B8004C | MOV AX,4C00 | 09BD | 0000 | 0000 | 0000 | 0100 | 48BD | 48AD | 48BF | 000C | 00000010 |
| 7 | CD 21 | INT 21 | 4C00 | 0000 | 0000 | 0000 | 0100 | 48BD | 48AD | 48BF | 000F | 00000010 |

Таблица трассировки программы z2

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Шаг | Машинный код | Команда | Регистры | | | | | | | | |  |
| AX | BX | CX | DX | SP | DS | SS | CS | IP | CZSOPAID |
| 1 | BDAF48 | MOV AX, 48AF | 0000 | 0000 | 0000 | 0000 | 0100 | 489D | 48B1 | 48AD | 0000 | 00000010 |
| 2 | 8ED8 | MOV DS, AX | 48AF | 0000 | 0000 | 0000 | 0100 | 489D | 48AD | 48BF | 0003 | 00000010 |
| 3 | B409 | MOV AH, 09 | 48AF | 0000 | 0000 | 0000 | 0100 | 48AF | 48AD | 48BF | 0005 | 00000010 |
| 4 | BA0000 | MOV DX, 0000 | 09AF | 0000 | 0000 | 0000 | 0100 | 48AF | 48AD | 48BF | 0007 | 00000010 |
| 5 | CD21 | INT 21 | 48AF | 0000 | 0000 | 0000 | 0100 | 48AF | 48AD | 48BF | 000A | 00000010 |
| 6 | B8004C | MOV AX,4C00 | 48AF | 0000 | 0000 | 0000 | 0100 | 48AF | 48AD | 48BF | 000C | 00000010 |
| 7 | CD 21 | INT 21 | 4C00 | 0000 | 0000 | 0000 | 0100 | 48AF | 48AD | 48BF | 000F | 00000010 |

Таблица трассировки программы z3

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Шаг | Машинный код | Команда | Регистры | | | | | | | | |  |
| AX | BX | CX | DX | SP | DS | SS | CS | IP | CZSOPAID |
| 1 | B409 | MOV AH, 09 | 0000 | 0000 | 0000 | 0000 | FFFE | 489D | 489D | 489D | 0100 | 00000010 |
| 2 | BA0C01 | MOV DX, 010C | 0900 | 0000 | 0000 | 0000 | FFFE | 489D | 489D | 489D | 0102 | 00000010 |
| 3 | CD21 | INT 21 | 0900 | 0000 | 0000 | 010C | FFFE | 489D | 489D | 489D | 0105 | 00000010 |
| 4 | 53 | MOV AX, 4C00 | 0900 | 0000 | 0000 | 0000 | FFFE | 489D | 489D | 489D | 0107 | 00000010 |
| 5 | CD21 | PUSH BX | 4C00 | 0000 | 0000 | 0000 | FFFE | 489D | 489D | 489D | 000A | 00000010 |

Ответы на контрольные вопросы

1. Сегментный (базовый) адрес - это адрес начала сегмента в памяти, который используется в сегментной адресации для доступа к данным и коду программы.

2. SS – 100h, DS – 15h, CS – 11h, Базовые адреса сегментов: SS – 48ADh, DS – 489Dh, CS – 48BFh. (48ADh\*10h + 100h) / 10h = 489Dh, (489Dh\*10h + 15h) / 10h = 48BFh.

3. Потому что физический адрес сегментов должен быть кратен 16.

4. MOV AX, 48BD – B8BD48, в данном случае 48BD - это машинное слово, которое имеет размер 2 байта. MOV AH, 09 – B409, в данном случае 09 занимает лишь байт. Размер 1ой команды - 3 байта, а второй - 2 байта, т.к в первом случае помещается машинное слово, а во втором - байт.

5. DS – 48AF, SS – 48B1, CS – 48AD. Сначала идет CS, потом идет SS и потом DS.

6. Во второй - 0100h, в третьей - FFFEh. Это происходит потому, что в третьей программе используется модель памяти TINY, в которой код, данные и стек находятся в одном сегменте.

7. Директивы ассемблера - это команды, которые управляют процессом ассемблирования и описывают структуру программы. Некоторые примеры директив включают ORG (организация памяти), SEGMENT (определение сегментов), DB (определение байтовых данных) и DW (определение слов).

8. Метка, помечающая первую команду программы в end, используется для указания точки завершения программы. Она помогает ассемблеру понять, где заканчивается код программы и начинаются другие данные или инструкции.

9. Числа размером в слово (16 бит) хранятся в памяти как последовательность двух байтов. Для загрузки значений в 16-битные регистры (например, AX) используется инструкция MOV.

10. Для инициализации текстовых строк в программе, строки могут быть объявлены с использованием директивы DB или DW.

11. Для вывода строки на экран с использованием DOS, необходимо использовать прерывание DOS, например, INT 21h. Строка передается в регистре DX, и DOS определит конец строки на основе символа нулевого байта (символ конца строки).

12. EXE-файлы могут иметь различные сегменты для кода и данных, в то время как .COM-файлы обычно имеют только один сегмент.