

## **ОТЧЕТ**

**За лабораторную работу № 3**

**Тема:**

**«Структура программы на Ассемблере.  
Вывод текстовой информации на монитор»**

**Дисциплина «Вычислительные машины и системы»**

Выполнил:

студент группы ИБ-92вп, Кадамов Ш.Ф.  
«\_» \_\_\_\_\_ 2022 г. \_\_\_\_\_ / Кадамов Ш.Ф.

Принял: П.А. Волынкин  
«\_» \_\_\_\_\_ 2022 г. \_\_\_\_\_ / П.А. Волынкин

## Цель

Целью настоящей работы является приобретение навыков работы:

- с простейшими командами ввода/вывода на уровне DOS;
- с системой программирования "Турбо Ассемблер";
- с отладчиком TD для ассемблера;

## Задача

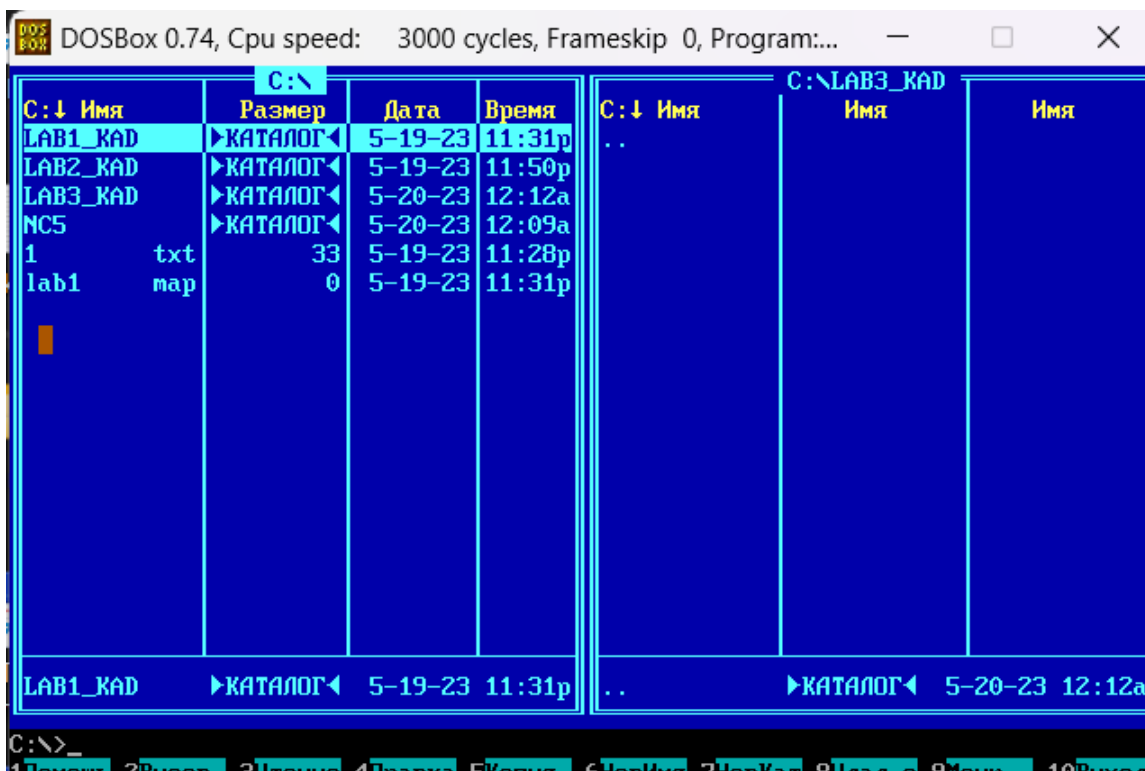
1. Запустить DOSBOX. Смонтировать виртуальный диск к папке C:\TASM
2. Войти в виртуальный диск.
3. Запустить NC
4. Создать файл с расширением asm
5. Набрать исходный текст программы (варианты 1 и 3 из п.3) во встроенном текстовом редакторе NC, сохранить в файле. Текст выводимого сообщения для каждой бригады должен быть свой – содержать имена и фамилии студентов в бригаде (латинскими и русскими буквами в кодировке 866), а также номер группы и номер бригады.
6. Произвести компиляцию и линковку программы.
  - 6.1. Компиляция  
`tasm *.asm à *.obj`
  - 6.2. Линковка (связывание)  
`tlink *.obj à *.exe`
7. Запустить программу на выполнение.
8. Написать программу вывода того же самого сообщения в консольном режиме на одном из языков высокого уровня (Delphi 1.0, Turbo Pascal, Turbo C, C++ и т.д.).
9. С помощью программы TD просмотреть и распечатать ассемблерный текст для всех программ. Разобраться с тем, в каком блоке интерфейса этой программы, какая содержится информация.
10. Сравнить результаты выполнения трех-четырех вариантов по размеру исходного и исполняемого файла. Сделать выводы.

11. По исходному коду и информации от TD определить:

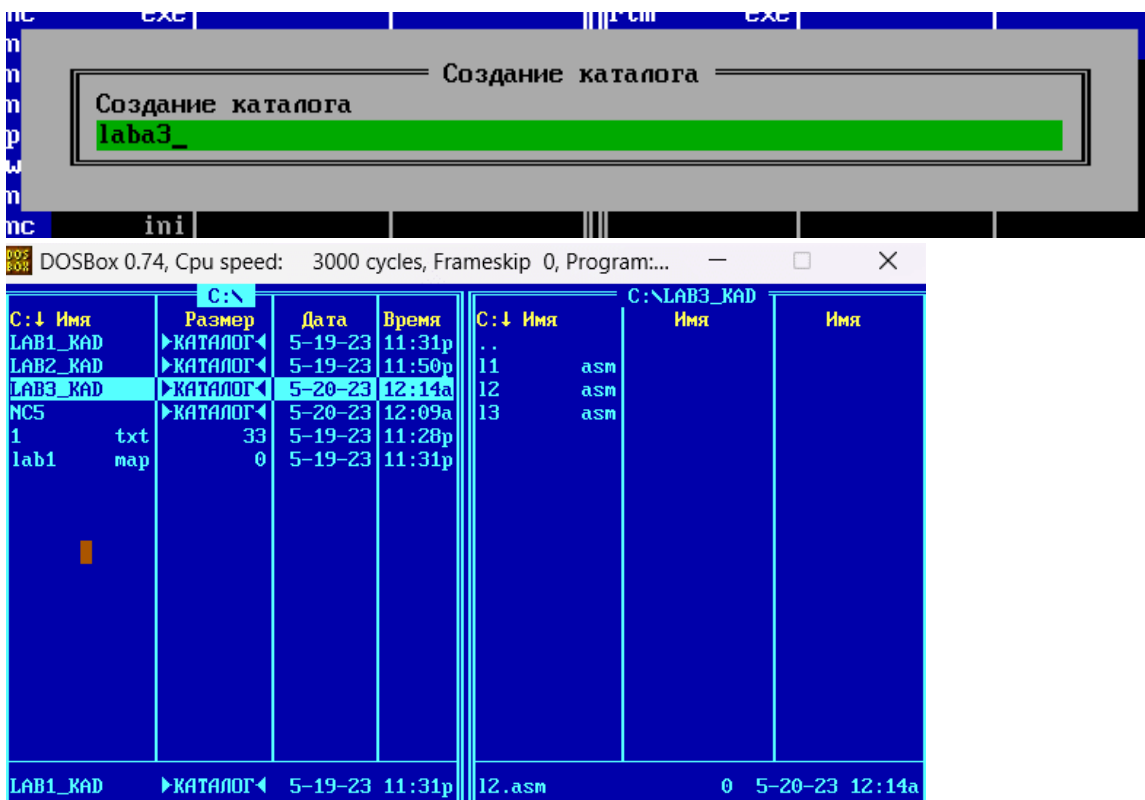
- Размеры всех трех сегментов и PSP
- Начальные и конечные адреса всех трех сегментов, а также префикса программы.
- Информацию по начальным, конечным адресам и размеру всех сегментов и PSP представить как в табличном, так и графическом вариантах.

## Ход работы:

1. Запущена виртуальная машина. Смонтирована рабочая папка. Запущен NC.



2. Используя команду F7 создали каталог, в котором используя команду SHIFT+F4 созданы файлы с расширением asm



### 3. Используя редактор NC и команду F4 введен исходный код в три файла.

- В файл L1.asm введен код общего файла Eхе.

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program:...
Файл: C:\lab3_kad\L1.asm * Строка 6 Столбец 6 63,475 свободно 115
SSEG segment stack ;Сегмент стека
db 256 dup(0)
SSEG ends

DSEG segment ;Сегмент данных
mess db 'ИБ92-вп, Кадамов Шерзод' ; Текст сообщения для вывода на экран
DSEG ends

CSEG segment ;Сегмент кода
assume cs:CSEG,ds:DSEG,ss:SSEG
begin:
mov ax, DSEG ; через регистр ax в регистр dx засылается начальный
mov ds,ax ; адрес сегмента данных

mov ah,09h ; согласно протоколу в регистр ah засылается функция
; вывода строки на экран до '$'
mov dx,offset mess ; теперь сообщение начинается с адреса ds:dx
int 21h ; согласно протоколу с адреса ds:dx будет выводится
; последовательность байт до тех пор пока не встретится
; символ '$'
mov ah,01h ; организация точки останова
int 21h ; (программа прекращает выполнение кода до нажатия
; любой клавиши)

mov ah,4ch ; обработка окончания
int 21h ; программы
CSEG ends ; завершение сегмента кода
end begin ; Завершение программы
```

#### Код файла L1:

```
SSEG segment stack ;Сегмент стека
db 256 dup(0)
SSEG ends

DSEG segment ;Сегмент данных
mess 'db ИБ92-вп, бригада 3: Блинов Александр, Горелова Валерия, Медведев Михаил','$' ;
Текст сообщения для вывода на экран
DSEG ends

CSEG segment ;Сегмент кода
assume cs:CSEG,ds:DSEG,ss:SSEG
begin:
mov ax, DSEG ; через регистр ax в регистр dx засылается начальный
mov ds,ax ; адрес сегмента данных

mov ah,09h ; согласно протоколу в регистр ah засылается функция
; вывода строки на экран до '$'
mov dx,offset mess ; теперь сообщение начинается с адреса ds:dx
int 21h ; согласно протоколу с адреса ds:dx будет выводится
; последовательность байт до тех пор пока не встретится
; символ '$'
mov ah,01h ; организация точки останова
int 21h ; (программа прекращает выполнение кода до нажатия
; любой клавиши)

mov ah,4ch ; обработка окончания
int 21h ; программы
CSEG ends ; завершение сегмента кода
end begin ; Завершение программы
```

- В файл L2.asm введен код упрощенного варианта файла .exe на основе модели памяти small.

```

Текст: C:\nc\Iaba3\I2.asm          Столбец 0      612 байт      0%
.8086                               ; Набор команд для процессора не ниже
8086
.MODEL small                         ; Модель памяти
.stack 100h                          ; Размер стека 256 байт
.data                                 ; Начало сегмента данных
    mess db "ИБ92-вп, Бригада 3: Блинов Александр, Горелова Валерия, Медведев
Михаил '$'                          ; Текст сообщения для вывода на экран
.code                                 ; Начало сегмента кода
start:
    mov dx,@stack                    ; через регистр dx в стековый
    mov ss,dx                        ; регистр ss засылается
                                       ; начальный адрес сегмента стека

    mov dx,@data
    mov ds,dx

    mov dx, offset mess
    mov ah,09h
    int 21h

    mov ah,01h
    int 21h

1Помощь 2>> 3 4Коды 5 6 7Поиск 8Формат 9 10Выход

```

### Код файла L2:

```

.8086                               ; Набор команд для процессора не ниже 8086
.MODEL small                         ; Модель памяти
.stack 100h                          ; Размер стека 256 байт
.data                                 ; Начало сегмента данных
    mess db 'ИБ92-вп, бригада 3: Блинов Александр, Горелова Валерия, Медведев Михаил','$'
                                       ; Текст сообщения для вывода на экран
.code                                 ; Начало сегмента кода
start:
    mov dx,@stack                    ; через регистр dx в стековый регистр ss
    mov ss,dx                        ; засылается
                                       ; начальный адрес сегмента стека

    mov dx,@data
    mov ds,dx

    mov dx, offset mess
    mov ah,09h
    int 21h

    mov ah,01h
    int 21h

    mov ah,4Ch
    int 21h

end start

```

- В файла L3.asm введен код варианта com программы.

```

екст: C:\nc\laba3\L3.asm          Столбец 0          701 байт          100%
; программа .com выводит на экран сообщение Privet
; и завершается после нажатия Enter

.model tiny ; Модель памяти для COM
.code      ; Начало сегмента кода
.org 100h  ; Начальное значение счетчика - 100h
start:
mov ah,09
mov dx,offset mess
int 21h
mov ah,01h
int 21h
ret ; Завершение COM-программы
mess db "ИБ92-вп, Бригада 3: Блинов Александр, Горелова Валерия, Медведев
Михаил '$' ; Строка для вывода
; с возвратом каретки, переводом строки,
; и признаком конца строки
end start ; Конец программы

```

### Код файла L3:

; программа .com выводит на экран сообщение Privet  
; и завершается после нажатия Enter

```

.model tiny ; Модель памяти для COM
.code      ; Начало сегмента кода
.org 100h  ; Начальное значение счетчика - 100h
start:
mov ah,09
mov dx,offset mess
int 21h
mov ah,01h
int 21h
ret ; Завершение COM-программы
mess db "ИБ92-вп, бригада 3: Блинов Александр, Горелова Валерия, Медведев Михаил','$' ;
Строка для вывода
; с возвратом каретки, переводом строки,
; и признаком конца строки
end start ; Конец программы

```

4. Используя команду TASM и TLINK, выполнена компиляция и линковка файлов L1.asm, L2.asm, L3.asm.

- Компиляция и линковка файла L1

```
C:\NC\LABA3>TASM L1
Turbo Assembler Version 4.1 Copyright (c) 1988, 1996 Borland International

Assembling file:  L1.ASM
*Warning* L1.ASM(6) Missing end quote
Error messages:   None
Warning messages: 1
Passes:           1
Remaining memory: 451k

C:\NC\LABA3>TLINK L1
Turbo Link Version 7.1.30.1. Copyright (c) 1987, 1996 Borland International
```

- Компиляция и линковка файла L2

```
C:\NC\LABA3>TASM L2
Turbo Assembler Version 4.1 Copyright (c) 1988, 1996 Borland International

Assembling file:  L2.ASM
*Warning* L2.ASM(5) Missing end quote
Error messages:   None
Warning messages: 1
Passes:           1
Remaining memory: 450k

C:\NC\LABA3>TLINK L2
Turbo Link Version 7.1.30.1. Copyright (c) 1987, 1996 Borland International
```

- Компиляция и линковка файла L3

Для линковки файла com требуется использование параметра /t

```
C:\NC\LABA3>TASM L3
Turbo Assembler Version 4.1 Copyright (c) 1988, 1996 Borland International

Assembling file:  L3.ASM
*Warning* L3.ASM(14) Missing end quote
Error messages:   None
Warning messages: 1
Passes:           1
Remaining memory: 451k

C:\NC\LABA3>TLINK /t L3
Turbo Link Version 7.1.30.1. Copyright (c) 1987, 1996 Borland International
```



## 5. Запущены созданные программы.

```
C:\NC\ЛАВА3>L1
ИБ92-вп, бригада 3: Блинов Александр, Горелова Валерия, Медведев Михаил '
C:\NC\ЛАВА3>L2
ИБ92-вп, Бригада 3: Блинов Александр, Горелова Валерия, Медведев Михаил '
C:\NC\ЛАВА3>L3
ИБ92-вп, Бригада 3: Блинов Александр, Горелова Валерия, Медведев Михаил '

```

## 6. Программа вывода того же самого сообщения в консольном режиме на языке C++

```
#include <iostream>

int main()
{
    setlocale(LC_ALL, "russian");
    std::cout << "ИБ92-вп, Бригада №3: Блинов Александр, Горелова Валерия, Медведев Михаил";
}

```

Консоль отладки Microsoft Visual Studio

```
ИБ92-вп, Бригада №3: Блинов Александр, Горелова Валерия, Медведев Михаил

```

Код программы на C++:

```
#include <iostream>
```

```
int main()
```

```
{
```

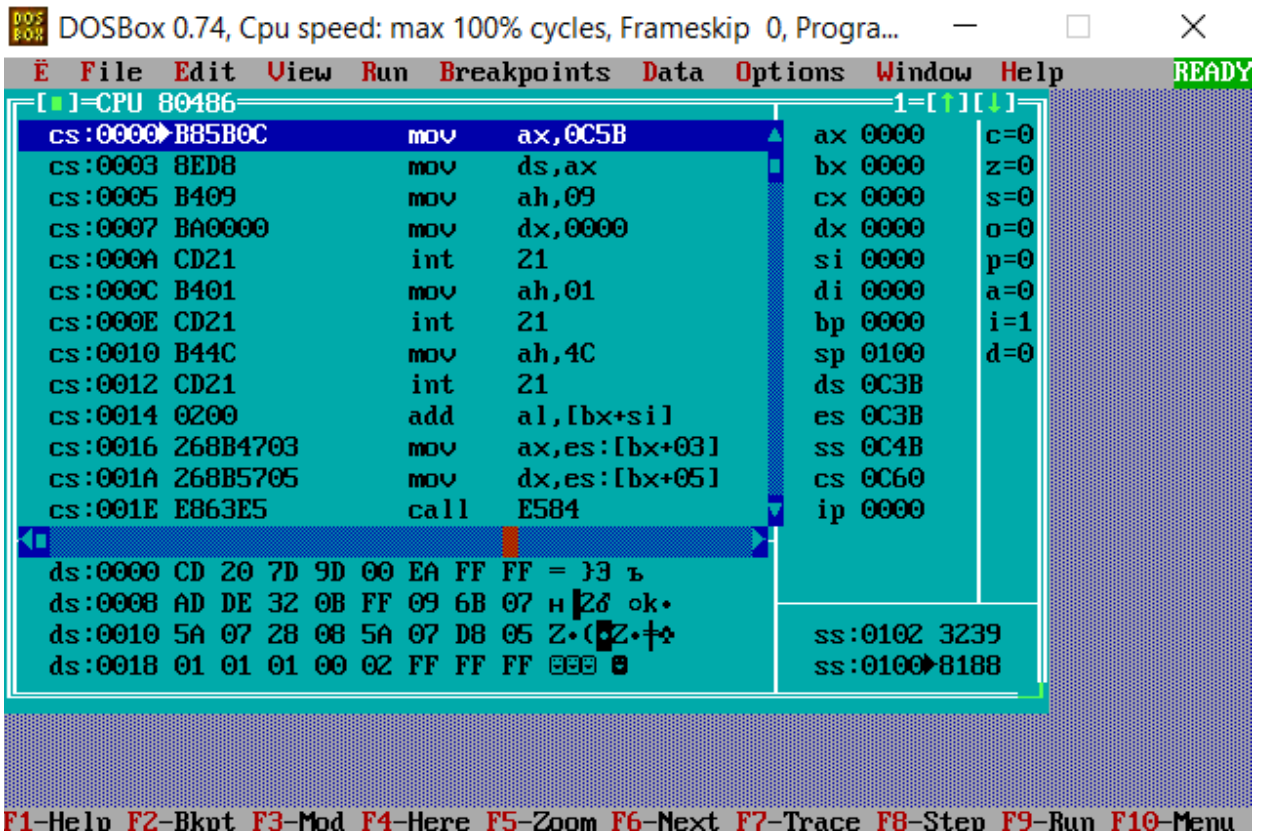
```
    setlocale(LC_ALL, "russian");
```

```
    std::cout << "ИБ92-вп, Бригада №3: Блинов Александр, Горелова Валерия,  
Медведев Михаил";
```

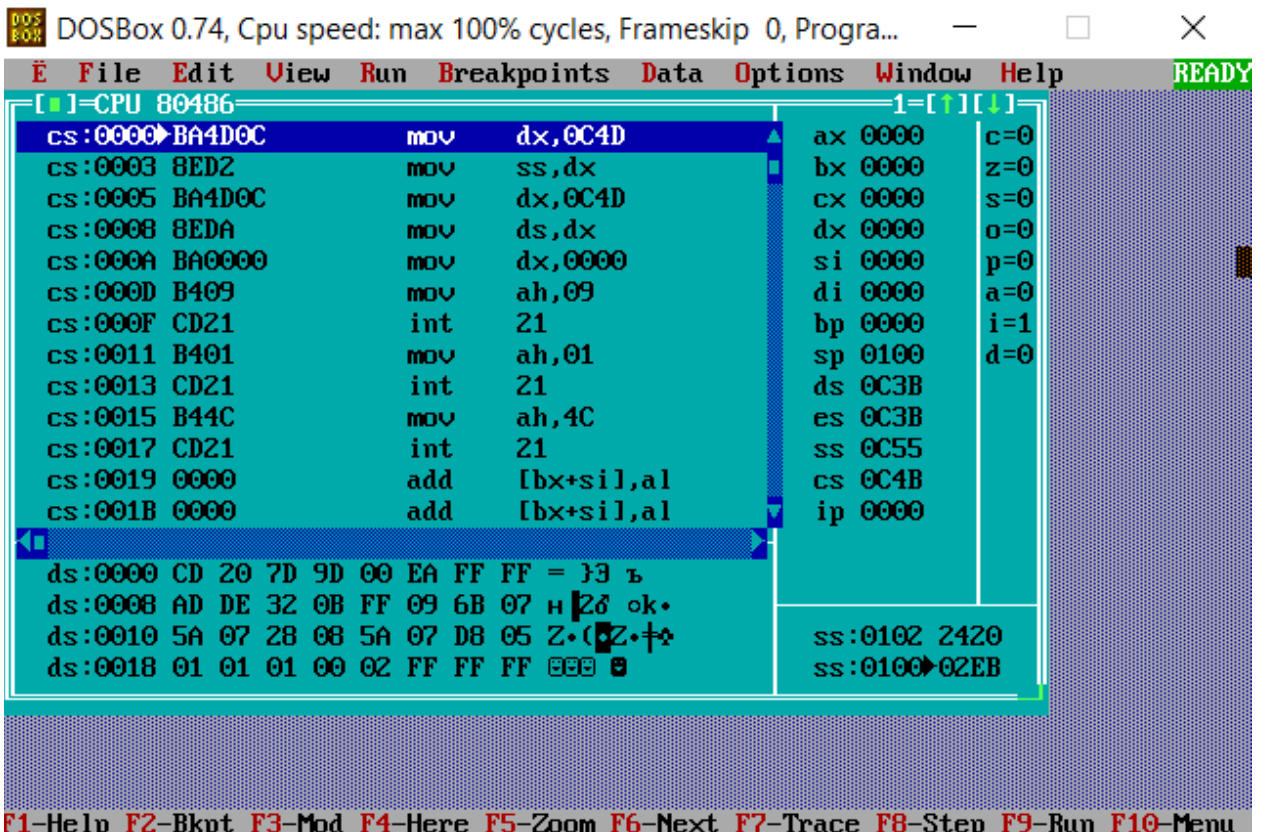
```
}
```

7. С помощью программы Turbo debugger (TD) просмотрен ассемблерный текст для всех программ. Используя команду TD «Имя программы».

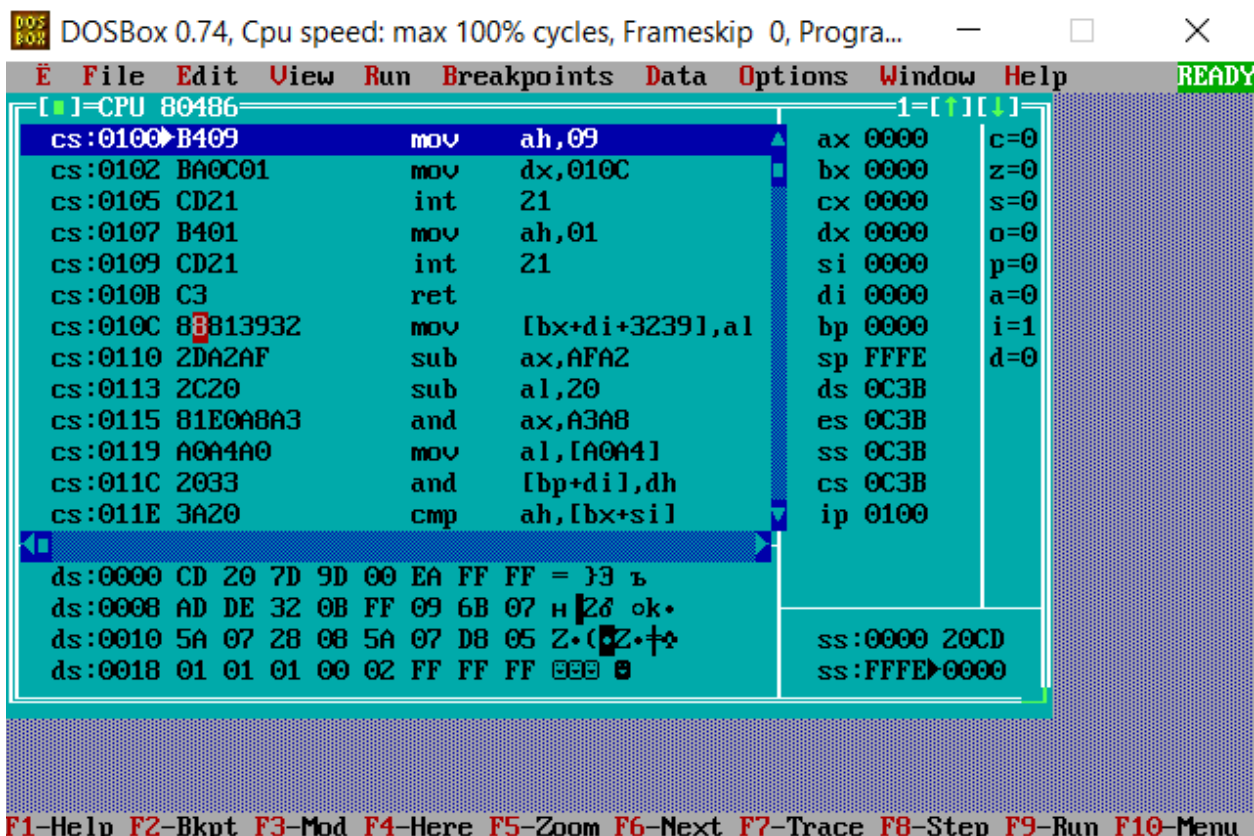
- Просмотрен текст L1



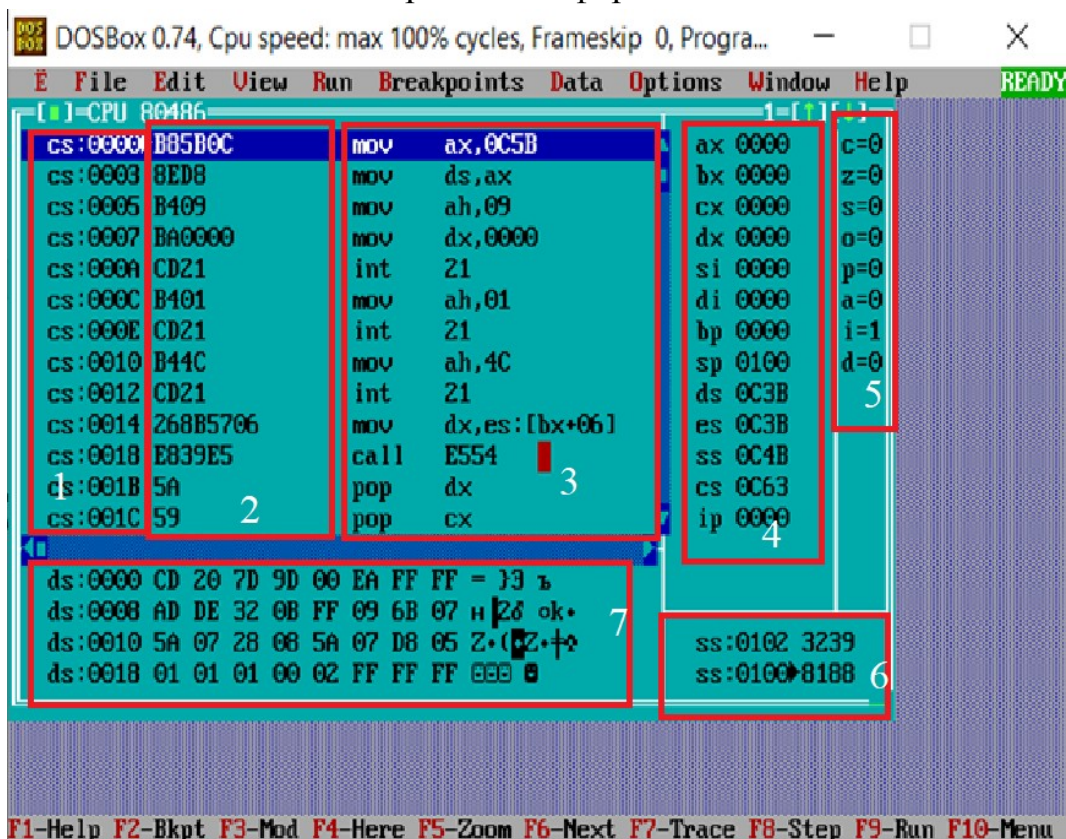
- Просмотрен текст L2



- Просмотрен текст L3



7.1. Разобраться с тем, в каком блоке интерфейса этой программы, какая содержится информация:



1. Адрес (Сегмент:смещение)
2. Машинный код
3. Ассемблерный код.

Для нас же главными здесь являются сегменты, а именно 3 сегмента и 1 префикс:

- cs – сегмент кода;
- ds – сегмент данных;
- ss – сегмент стека.
- PSP – префикс программного сегмента (структура данных)

4. Окно регистров (ax, bx, cx...).
5. Окно флагов.
6. Окно стека
7. Данные программы

Теперь кратко о том, что происходит, если мы будем пошагово выполнять команды с помощью горячей клавиши F8.

Например, регистру “ax” даётся значение, которое указано в ассемблерном коде, а у регистра “ip” изменилось значение на размер выполненной машинной команды (смещение).

Также регистру “ds” даётся новое значение, которое тоже взято из ассемблерного кода. В последствии будет ещё так же изменён и регистр “dx”.

После определённого количества нажатий на F8, то есть выполнения команд выводится то, что мы уже не раз видели, а именно, наш текст.

## 8. Сравнение результатов выполнения 3-х программ.

	Исходный файл (байт)	Исполняемый файл (байт)	Процентное соотношение
Файл L1	1214	916	23%
Файл L2	611	658	16,9%
Файл L3	701	110	100%

Из таблицы видно, что конечные файлы L1 и L3 программ на языке ассемблер размер меньше, чем исходный код, причем программа типа

COM значительно меньше. А конечный файл L2 получился немного больше исходного.

## 9. Табличный вид:

### 9.1. Программа с кодом общего файла Exe

			Файл L1
	Начало	Размер (байт)	Конец
PSP	0C3B0h	256	0C4B0h
Сегмент стека	0C4B0h	256	0C5B0h
Сегмент данных	0C5B0h	72	0C600h
Сегмент кода	0C600h	20	0C644h

### Графический вид L1.

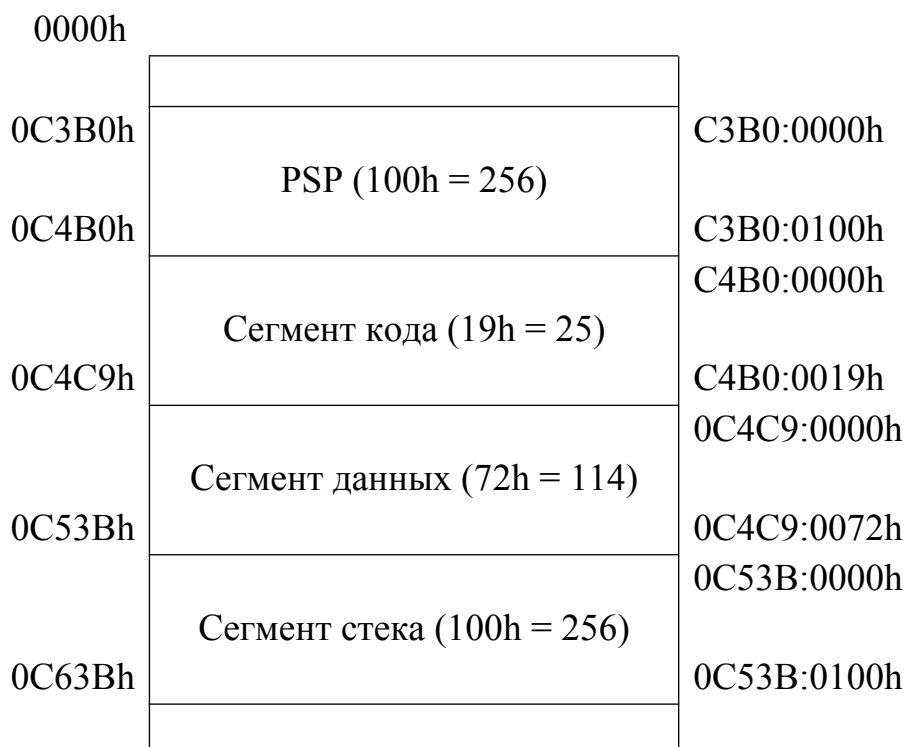
0000h		
0C3B0h	PSP (100h = 256)	0C3B0:0000h
0C4B0h		0C3B0:0100h
	Сегмент стека (100h = 256)	0C4B0:0000h
0C5B0h		0C4B0:0100h
	Сегмент данных (71h = 113)	0C5B0:0000h
0C621h		0C5B0:0071h
	15 байт	
0C630h	Сегмент кода (14h = 20)	0C630:0000h
0C644h		0C630:0014h

9.2. Программа с кодом упрощенного варианта файла .exe на основе модели памяти small.

Файл L2

	Начало	Размер (байт)	Конец
PSP	0C3B0h	256	0C4B0h
Сегмент кода	0C4B0h	25	0C4C9h
Сегмент данных	0C4C9h	114	0C53Bh
Сегмент стека	0C53Bh	256	0C63Bh

Графический вид L2.

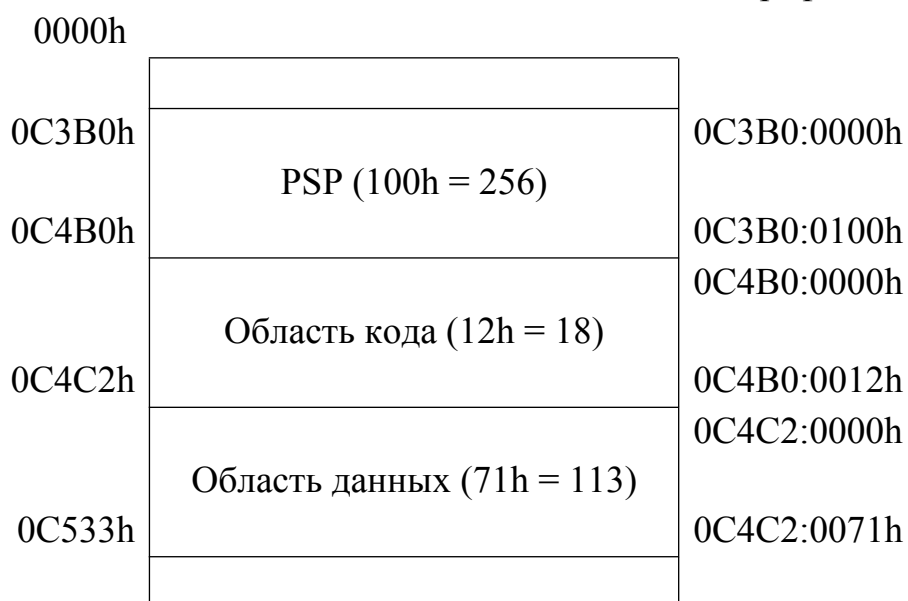


### 9.3. Программа с кодом варианта com программы.

Файл L3

	Начало	Размер (байт)	Конец
PSP	0C3B0h	256	0C4B0h
Область кода	0C4B0h	18	0C4C2h
Область данных	0C4C2h	113	0C533h
Область стека	-	-	-

Графический вид L3.



## **Вывод:**

В ходе данной лабораторной работы мы, 3-я бригада, изучили основы написания кода, компиляцию и линковку программ на базе Турбо Ассемблер. Используя программу Turbo debugger научились определять адреса сегментов памяти, их размер и порядок.

Для достижения столь недостижимой цели, нам потребовалось создать файлы с расширением .ASM. Были сравнены несколько разных типов приложений на Ассемблере, тип Eхе больше возможностей, но занимает больше места на диске чем файл COM.

Файл COM имеет ограничение в размер одного стека в 64кБайт. В COM файле данные располагаются в сегменте кода, а не в сегменте данных.