###### tlЛабораторная работа №1

**Освоение инструментальной среды для выполнения лабораторных работ (DosBox, Tasm, Tlink, TurboDebugger)**

**DosBox** – приложение имитирующее работу 16-разрядной операционной системы (DOS) от Microsoft. Из нее вы будете запускать:

* **Tasm** – компилятор языка Assembler (получите файл с рсширением .obj);
* **Tlink** – компоновщик, который позволяет получать исполняемый файл (.ехе);
* **TurboDebugger (TD)** – отладчик исполняемых файлов.

Последовательность действий при выполнении работы следующий:

1. Текстовый редактор (Notepad, Notepad++, Notepad2); набор текста программы; сохранение в файле в рабочем каталоге.
2. Запуск **Tasm** из **Dosbox**; трансляция (компиляция); если обнаружены синтаксические ошибки, то их анализ, исправление и возврат к п.1 на исправление иначе к п.3.
3. Запуск **Tlink**; компоновка; если обнаружены ошибки компоновки, то возврат к п.1 и исправление иначе к п.4.
4. Запуск **TD**; ввод данных; выполнение программы в отладчике; просмотр результатов, скриншоты для отчета.

**Подготовка рабочего каталога**

1. Создать (если нет) на локальном диске **D**: компьютера, в папке **Users** папку своей группы D:\Users\alocs\(№группы). Например D:\Users\alocs\21ВВ1 (**использовать только латиницу)**.

3. Скопировать с сетевого диска Z: (\\master.bt.local\Courses\alocs) в эту папку - папки «Литература» и "Методическое обеспечение". Если лабораторные работы выполняются ***не в лабораториях кафедры*** то нужно скачать архив по ссылке:

<https://cloud.mail.ru/public/EAfq/xRNnaZUHd>.

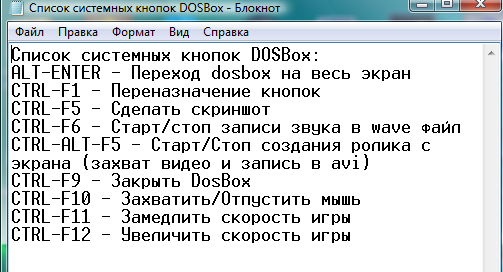
Затем разархивировать на локальный диск.

4. Ознакомиться с составом лабораторных работ.

З**апуск приложения DOSBOX**

5. Открыть папку D:\Users\alocs\(№ группы)\dosbox.

6. Изучить содержимое файла «Список системных кнопок DosBox.txt», открыв его по F4.

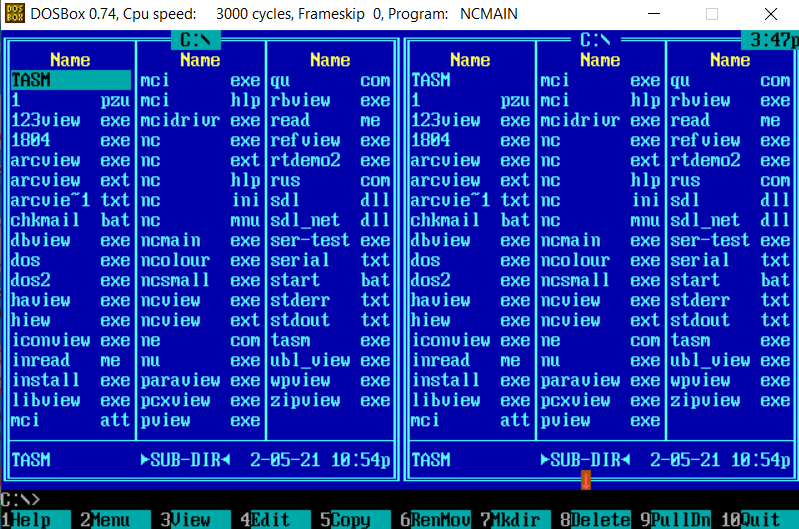


**Внимание:** захватить/отпустить мышь в Dosbox по Ctrl+F10;

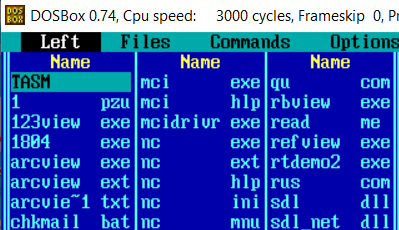
по Alt+Tab можно перейти в нужное приложение в Windows.

7. Запустить инструментальную среду «DOSBOX». Для этого в папке D:\Users\alocs\\dosbox,дважды щелкнуть (запустить) командный файл «NC.bat».

Должно получиться:



8. В результате будет открыто приложение "DOSBOX", которое позволяет запускать 16-разрядные приложения (**Tasm, Tlink, TD**) на ОС: Windows 7, 8, 10 и файловый менеджер Norton Commander. Синее окно, которое вы видите это файловый менеджер Norton Commander (NC). С его помощью можно открывать и редактировать файлы, копировать файлы, создавать папки и т.д.



Далее открываем папку **TASM** (двойной левый клик). Эта папка будет **рабочей** для вас. Сюда нужно будет поместить файл с текстом программы на языке ассемблера.

**Подготовка исполняемого файла (exe-файла)**

Чтобы получить исполняемый exe-файл надо **текстовый** файл с программой на языке Ассемблер, находящийся в папке **TASM**:

* компилировать (перевести с Ассемблера в язык машинного кода)
* линковать (связать с библиотеками и разместить в оперативной памяти).

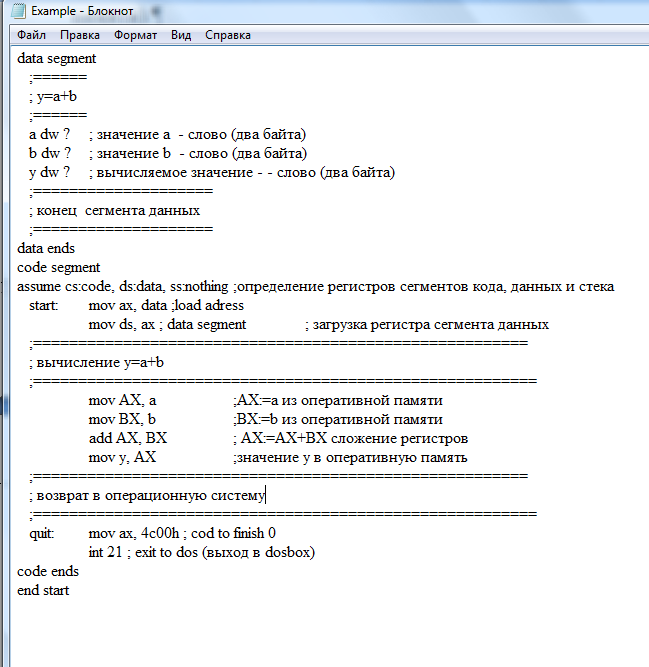
9. Рассмотрим это на примере файла **prim.asm** - программы на языке Ассемблер в папке TASM. Это файл с текстом программы на языке Ассемблер для процессора Intel 8086.

Эта программа вычисляет значение y=a+b.

Под переменные «**a»** и «**b»** отводится в оперативной памяти по одному слову (2 байта), т.е. 16 бит.

Результат «**y»** также слово.

Текст программы:



5

4

4

3

2

1

Блоки 1,3,5 будут в любой вашей программе на языке Ассемблера в таком же виде.

Блоки 2, 4 будут меняться в вашей программе (2 - какие переменные, 4 – что вычислять с их использованием).

10. Выполнить компиляцию этого файла.

Для этого в DOSBOX перейти в папку TASM и в самом низу NC (это командная строка) набрать выполнить (по **Enter**) команду компиляции: tasm.exe prim.asm

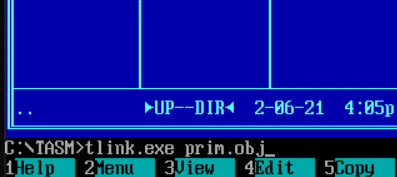


По комбинации клавиш <Ctrl+O> временно убрать окно NC (затем NC включается той же комбинацией клавиш).

Если все правильно сделано, то получите сообщение об отсутствии ошибок при компилировании и в папке появится файл prim.obj



10. Затем выполнить линкование задав команду tlink.exe prim.obj:



Если сделано все правильно, то убрав окно NC, увидите:



и в папке появится файл prim.exe

**Загрузка исполняемого файла (exe-файла)**

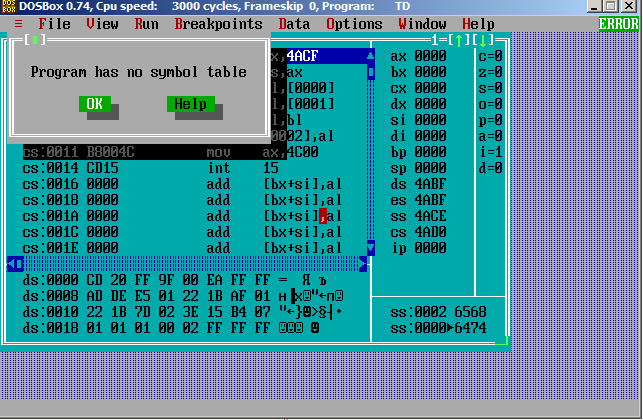
**с помощью отладчика TurboDebugger и выполнение программы**

Введите команду td.exe prim.exe:



Если сделано все правильно, то появится окно отладчика:

окно CPU



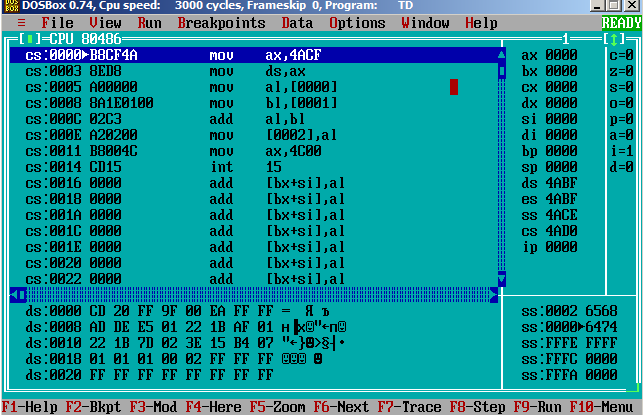
окно флагов

окно регистров

окно стека

окно ОП

Нажмите "ОК" и разверните отладчик на все окно, используя стрелку:



программа

11. Выполнить тестирование (прогон) программы для двух случаев:

1) y>0;

2) y<0.

Для этого подготовить свои значения a и b, аналогично примеру ниже:

1) a=20;b=30; y=50

2) a=20;b=-30; y=-10

Т.к. лучше работать с отладчиком в 16-чной системе переводим a, b ,y в эту систему:

1) a=14(16);b=1Е(16); y=32(16)

2) a=14(16);b=E2(16); y=F6(16)

Отрицательные числа в процессоре представляются в дополнительном коде, поэтому -30(10) будет равно (16-битовое):

-30(10)=-0000000000011110(2).

Инвертируем разряды двоичного числа:

1111111111100001.

Добавляем 1 в младший разрял:

1111111111100010.

Переводим в 16-чную систему:

FFE2(16)

Аналогично поступаем с -10(10).

Получаем:

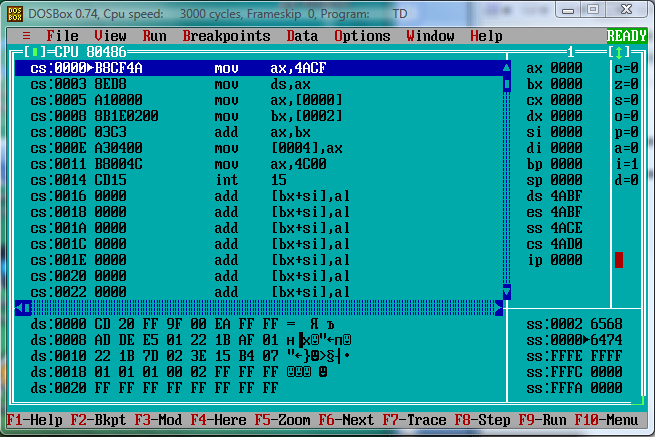
1) a=0014(16);b=001Е(16); y=0032(16)

2) a=0014(16);b=FFE2(16); y=FFF6(16)

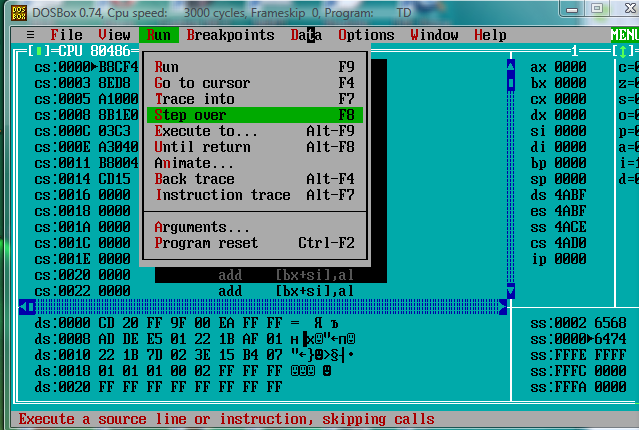
первая команда программы

Выполняем просчет для y>0.

Окно отладчика после загрузки программы:

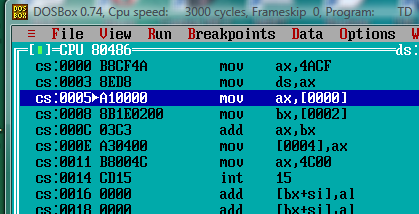


По нажатию клавиши F8 выполняем две команды (одно нажатие – одна команда):



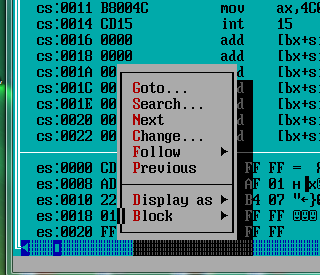
Получаем:

команда, которая будет выполняться следующей

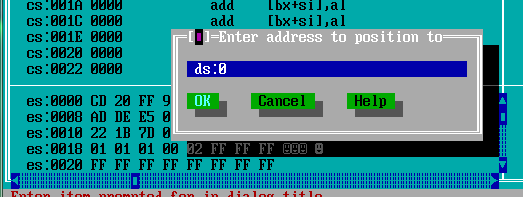


Эти первые две выполненные команды (mov ax,4ACF и mov ds,ax) устанавливают значение регистра сегмента данных DS (адрес оперативной памяти с которого располагаются переменные программы). Поэтому теперь можно задавать данные (a, b).

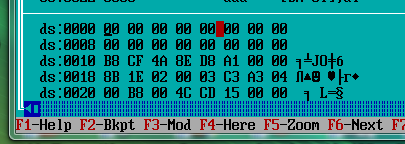
Переключаем в отладчике окно отображения оперативной памяти на начало сегмента данных. Правый клик на этом окне и выбираем **Goto**



Задаем начало сегмента данных **ds:0** и **ОК**

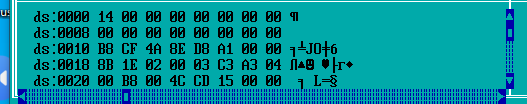


Получаем:

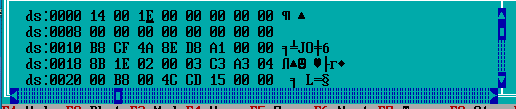


Задаем значение **а.** Правый клик на окне памяти, ставим курсор на адрес **0000**, выбираем **Change,** вводим **14** (с)**, ОК.** Ставим курсор на адрес **0001**, **Change,** вводим**, 00 ОК.**

Получаем значение а=0014:



Аналогично задаем значение b.



**Примечание:** ввод числа в 16-ой системе -> 0014h. Если число начинается с буквы, например АВ6, то 0АВ6h.

После ввода исходных данных, переходим в окно CPU и выполняем по F8 следующие четыре команды:

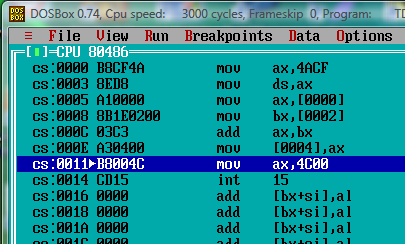
mov ax, [0000]-> смотрим, что АХ=0014

mov bx, [0002] ->смотрим, что ВХ=001Е

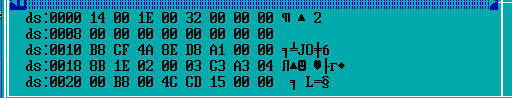
add ax, bx -> смотрим, что АХ=0032

mov [0004], ax-> смотрим, что в памяти по адресу 0004 записано значение 0032

Получаем:

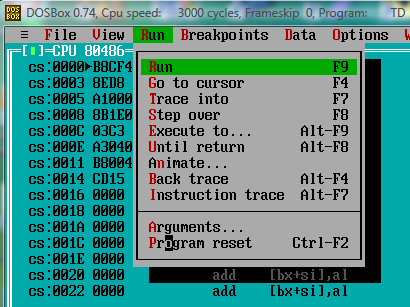


Переходим в окно оперативной памяти и смотрим значение **y**:

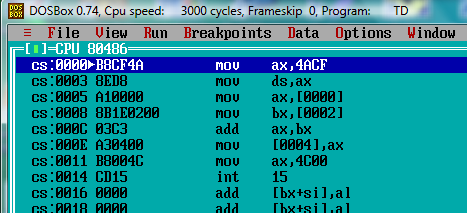


Все верно!

Чтобы выполнить второй просчет выполняем программный сброс (возврат на начало программы) по Ctrl +F2



Получаем:



Аналогично повторяя действия, вводим значения **a,b**. Выполняем программу, проверяем значение **у**.

Результат вычисления для **y<0** и **y>0** показать преподавателю.

**Корректное завершение работы:**

* выйти из отладчика: **File->Quit**;
* выйти из NC: по **F10**;
* выйти из DOSBOX по команде **exit**+**enter**

***Подготовить отчет по работе. Включить в него скриншоты подтверждающие вычисление выражения y=a+b для y>0 и y<0 для своих выбранных значений a и b.***