Министерство образования Республики Беларусь

Учреждение образования

Брестский государственный технический университет

Кафедра ИИТ

Лабораторная работа №1

За 4 семестр

По дисциплине «Компьютерные системы и сети»

Тема: «Язык ассемблера»

Выполнил: студент 2 курса

Группы ПО-4(2)

Коташевич С.Н.

Проверил: Савицкий Ю.В.

Брест 2021

**Лабораторная работа №1**

**Язык ассемблера**

**Часть 1:** Организация и исполнение программы на Ассемблере. Назначение и использование регистров МП. Изучение этапов создания программ на языке ассемблера.

**Цель:** Знакомство со структурой ассемблерной программы, процессом её получения и основными возможностями программ TASM, TLINK.

1. Пользуясь правилами оформления ассемблерных программ, набрать программу Hello в файл LAB01.ASM. Получить выполняемый файл и запустить его. Каков результат?

Код программы lab01.asm:

        .MODEL Tiny

        .STACK 100h

        .DATA

Msq     DB  'Hello.',13,10,'$'

        .CODE

        mov    ax, @Data

        mov    ds, ax

        mov    ah, 9

        mov    dx, OFFSET Msq

        int    21h

        mov    ax, 4C00h

        int    21h

        END

Получаем выполняемый файл и запускаем его:



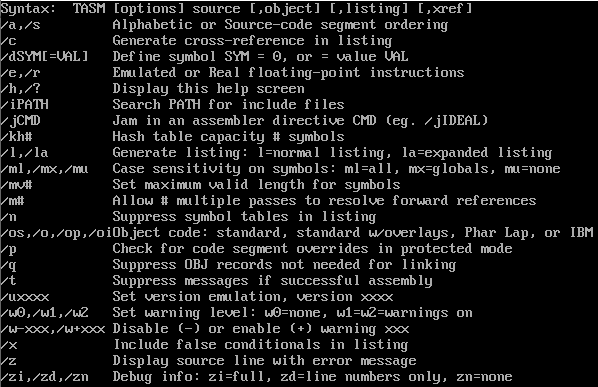
1. Определить, какие файлы создались при получении выполняемого файла. Пользуясь редактором, определить внутренний формат (текстовый или бинарный) КАЖДОГО из них.

Создались три файла:

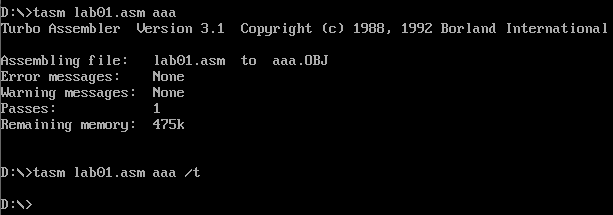
* LAB01.obj – текстовый файл
* LAB01.map – бинарный файл
* LAB01.exe – бинарный файл

1. Получить список ключей TASM, задав пустую командную строку.

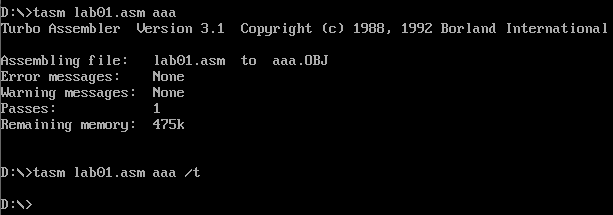
Вводим tasm. Получаем список ключей:



1. Получить файл AAA.OBJ из файла LAB01.ASM. Обратить внимание на сообщения TASM. Что изменилось? Выполнить трансляцию ещё раз с ключом /t. На что он влияет.

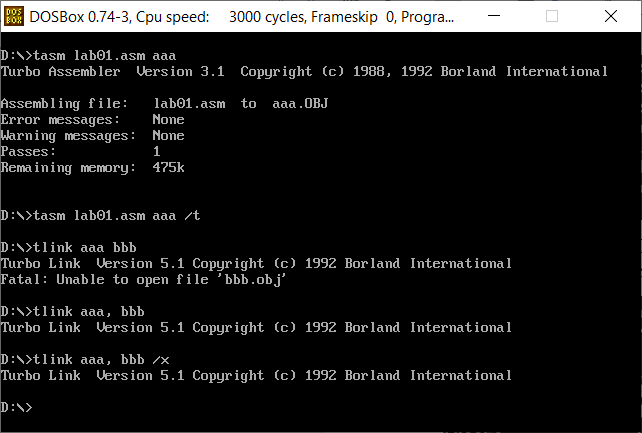


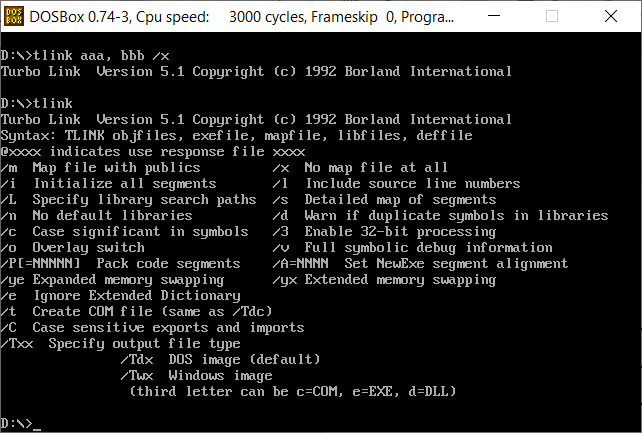
Имя ассемблируемого файла изменилось с lab01 на aaa.



Ключ /t скрывает появляющееся сообщение, если ассемблирование прошло успешно.

1. Получить из файла AAA.OBJ файл BBB.EXE без генерации карты загрузки. Догадайтесь, как это сделать, по аналогии с TASM. Каков вид командной строки?





**Часть 2:** Изучение и работа с основной частью файла листинга ассемблерной программы. Отладка программы с помощью программы Turbo Debugger.

**Цель:** Знакомство со структурой файла листинга, получаемого при ассемблировании, структурой машинных команд процессора; Изучение основных приёмов отладки программ и основных возможностей отладчика Turbo Debugger (TD).

* Изучить предложенный материал по структуре и назначению полей файла листинга ассемблерной программы.
* Изучить структуру машинной команды.

1. Пользуясь информацией лаб.раб.1, написать программу Lab2.asm, работающую по следующему алгоритму:

* вывести на экран первые N символов фамилии студента (использовать функцию 40h DOS INT 21h) – число N задаётся преподавателем;
* вывести строку полностью в цикле N-1 раз на экран.

Код программы lab02.asm:

        .MODEL Tiny

        .STACK 100h

        .DATA

Surname     DB  'Kotashevich',13,10,'$'

        .CODE

        mov    ax, @Data

        mov    ds, ax

        mov    ah, 40h

        mov    bx, 1

        mov    dx, OFFSET Surname

        mov    cx, 5

        int    21h

        mov    ah, 02h

        mov    dl, 10

        int    21h

        mov    ah, 9h

        mov    cx, 4

    mark:

        mov    dx, OFFSET Surname

        int    21h

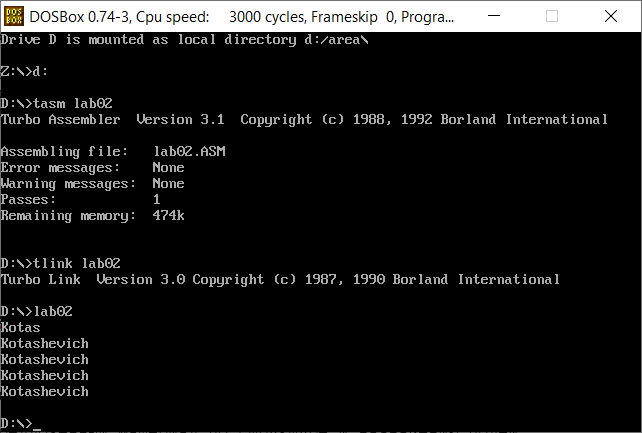
        loop   mark

        mov    ax, 4C00h

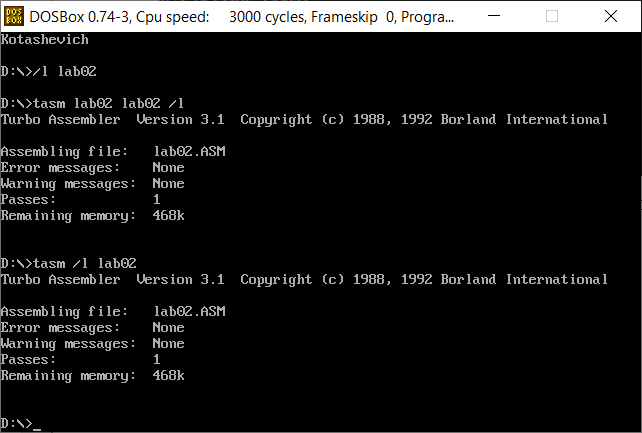
        int    21h

        END

Результат работы:

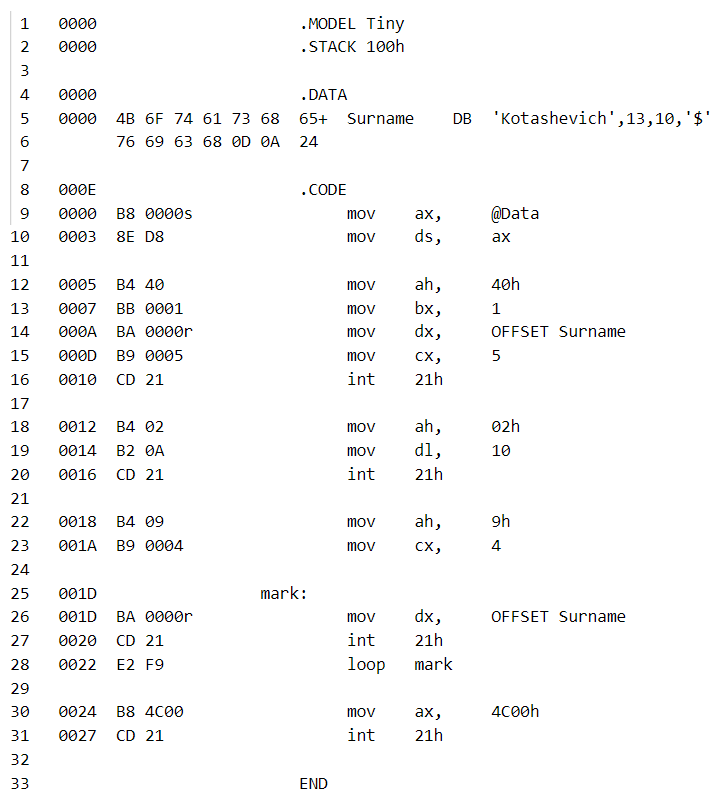
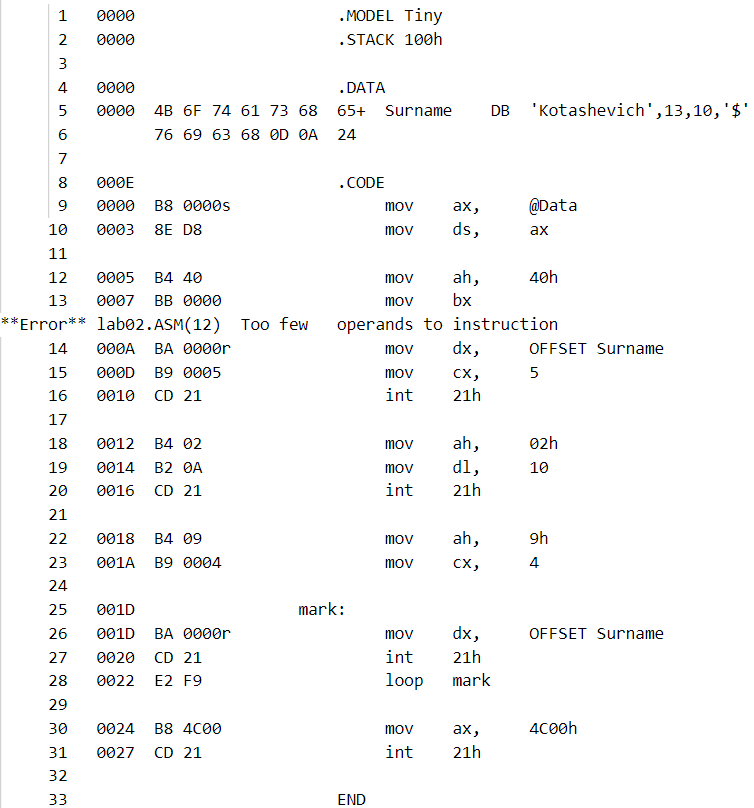


1. Получить файл листинга и внимательно ознакомиться с его форматом и содержимым.



Создался файл LAB02.LST

1. В любой инструкции с двумя операндами удалить один операнд и проассемблировать программу с получением файла листинга. Какие выходные файлы создаются в этом случае? Что добавляется в листинге?

**Привести результаты наблюдений в отчёте и объяснить их.**

Создаётся только файл LAB02.LST.

В листинге добавляется сообщение об ошибке: *Too few operands to instruction.*

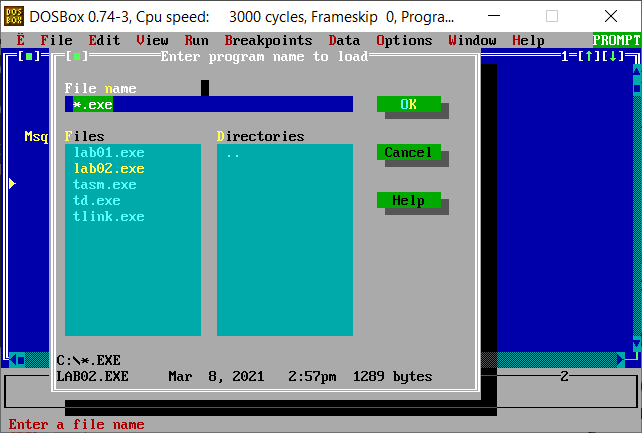
1. Восстановить исходную программу и создать загрузочный модуль Lab2.exe (при этом использовать команду d:\tasm\tasm /zi lab02.asm на этапе трансляции и команду lab2.obj d:\tasm\tlink /v lab02.obj на этапе компоновки. **Привести в отчёте назначение ключей /zi и /v**).

Ключ **/v** передаёт в загрузочный файл символьную информацию, позволяющую отладчику TD выводить на экран полный текст исходной программы, включая метки, комментарии и прочее.

Ключ **/zi** управляет включением в объектный файл номеров строк исходной программы и другой информации, не требуемой при выполнении программы, но используемой отладчиком.

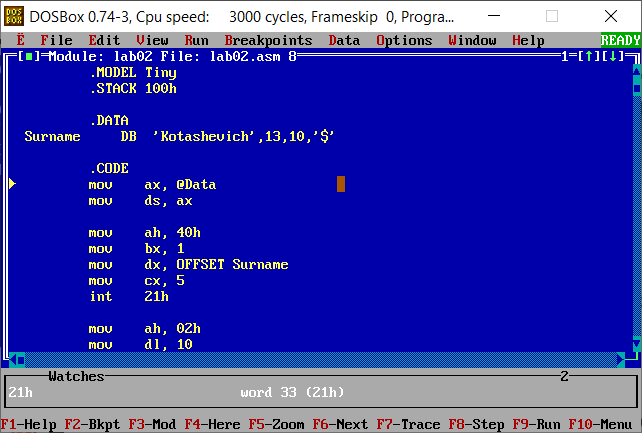
1. Загрузить программу в отладчик (td d:\asm\lab02). Какими способами это можно сделать? **Отразить способы в отчёте.**

Чтобы загрузить программу в TD, она должна быть предварительно оттранслирована ассемблером и преобразована в файл типа COM или EXE.

Заходим в меню FILE и выбираем пункт OPEN. На экране появится окно загрузки файла, содержащее в верхней части строку для ввода имени файла. Набираем в ней имя файла и нажимаем ENTER. 

(мб удалить скрин)

1. Вывести в окне Watch (View->Watch) содержимое выходного буфера.



1. Выполнить программу (Goto cursor – клавиша F4) до места начала цикла.
2. Выполнить 2 прохода цикла по F7, контролируя значения регистров. Для этого необходимо открыть окно CPU (View->CPU). Какие регистры изменяются в цикле.

Привести результаты выполнения данного фрагмента программы в отчёте в виде таблицы:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № шага | IP | AX | BX | CX | DX |
| 1 | 001D | 090A | 0001 | 0003 | 002A |
| 2 | 001D | 090A | 0001 | 0002 | 002A |

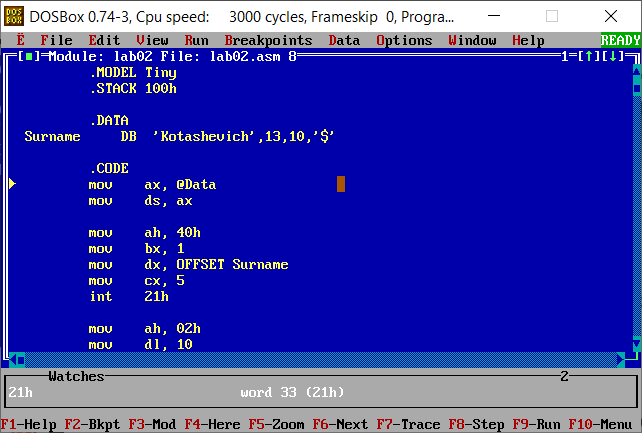
Изменилось значение регистра CX.

1. Остальные проходы цикла выполнить по F8. В чём разница?

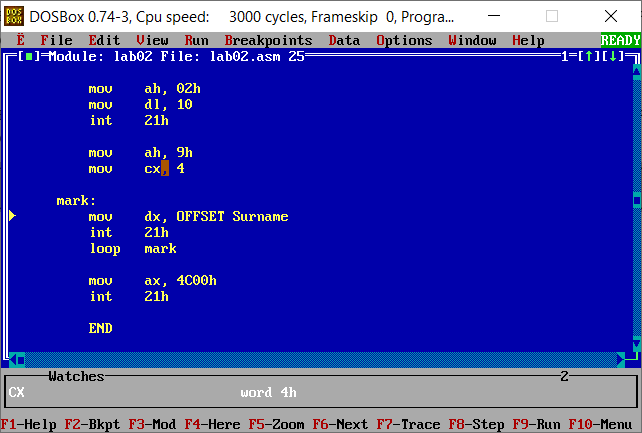
**Привести объяснения в отчёте.**

F8 отличается от F7 тем, что F8 проходит цикл не пошагово, а полностью.

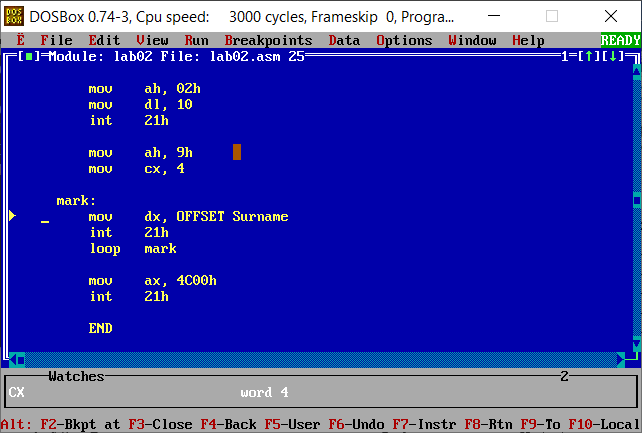
1. Вывести содержимое выходного буфера в окне Watches.



1. Вывести содержимое регистра СХ буфера в окне Watches в шестнадцатеричном формате (Options->Display options->Hex)



1. Вывести содержимое регистра СХ буфера в окне Watches в шестнадцатеричном формате (Options->Display options->Decimal)



1. Изменить содержимое регистра СХ на величину, меньшую на 2 единицы (Data->Evaluate/Modify->Expression=cx, New Value=cx-2). Выполнить пошагово (F7) программу до конца. Что изменилось? **Объяснить в отчёте.**

Уменьшив СХ на два, мы сокращаем количество итераций, выполняемых программой при выводе фамилии на 2.

1. Выполнить команду Animate из меню Run с задержкой 1000мс.

Программы выполняется с задержкой между инструкциями в 1000мс.

1. Сбросить программу по Ctrl-F2. Выполнить первые 2 инструкции. Отменить их действие по Alt-F4. **Привести объяснения о результате в отчёте.**

Вернулись начальные значения регистров. Инструкции можно отменять.

1. По шагам пройтись до первой после int 21h инструкции. Попытаться по Alt-F4 всё отменить. Что случилось? **Привести объяснения о результате в отчёте.**

Сочетание клавиш Alt-F4 включает функцию Backtrace, позволяющую идти по исполняемым инструкция в обратном порядке. При переходе на последующую инструкцию и нажатии Alt и F4 ничего не происходит, т.к. предыдущая инструкция int 21h является инструкцией прерывания.

1. Сбросить программу по Ctrl-F2. По F7 выполнить первые 6 инструкций. Открыть окно Execution History. Отменить последние 3 шага.

**Вывод:** ознакомился со структурой ассемблерной программы, процессом её получения и основными возможностями программ TASM, TLINK.