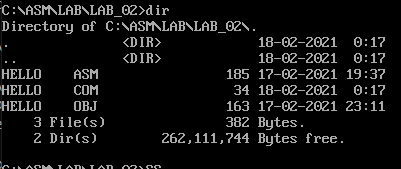
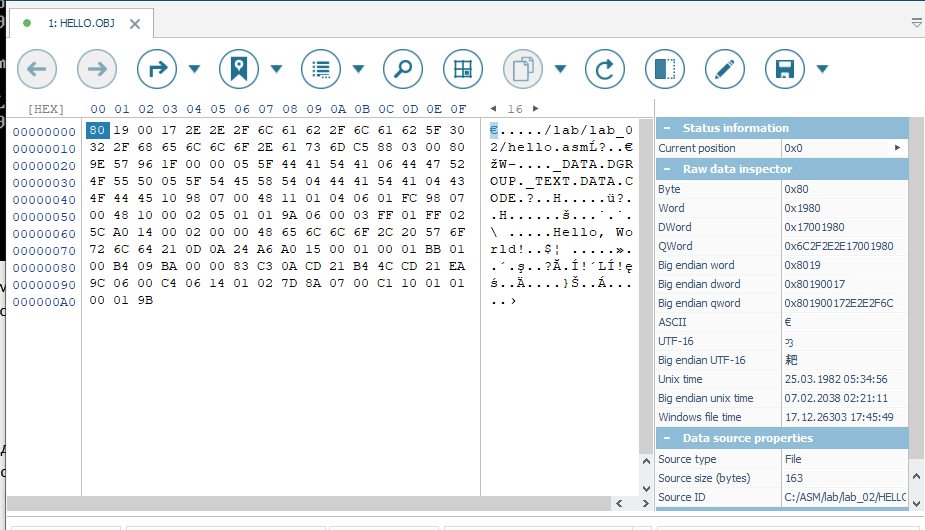
1. Создайте файл hello.asm с текстом .  
MODEL TINY   
.DOSSEG   
.DATA   
 MSG DB "Hello, World!", 0Dh, 0Ah, '$'   
.CODE   
.STARTUP   
 MOV BX, 1   
 MOV AH, 09h  
 MOV DX, OFFSET MSG   
 ADD BX, 10  
 INT 21h   
 MOV AH, 4Ch   
 INT 21h  
END

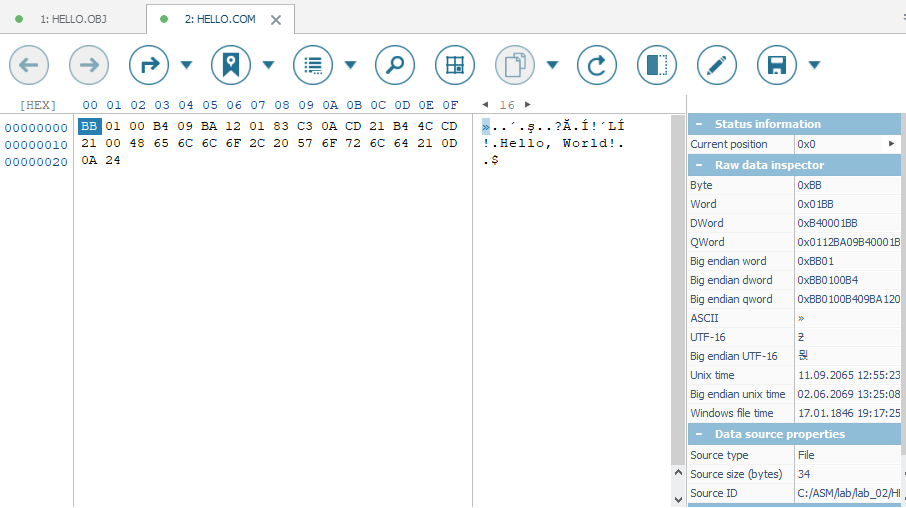
2. Запустите ML.EXE /AT hello.asm.



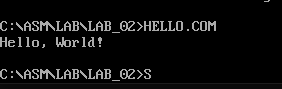
3. Посмотрите, какие файлы были созданы компилятором. Определите их размер. Просмотрите файл в любом hex-viewer’е, проанализируйте содержимое файла и изучите возможности просмотровщика.







4. Запустите скомпилированную программу.



5. Запустите AFDPRO.EXE HELLO.COM.

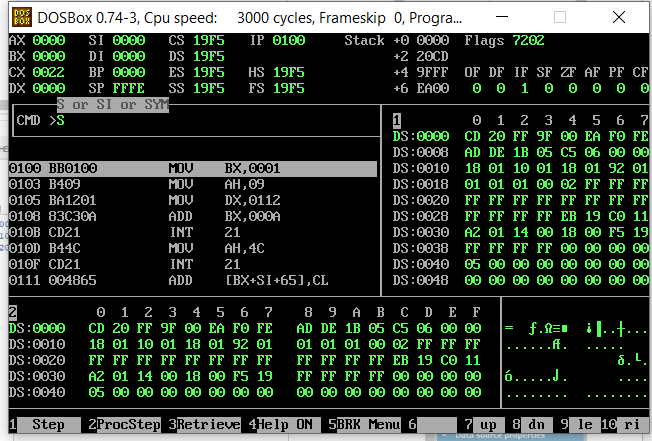
Верхний ряд:

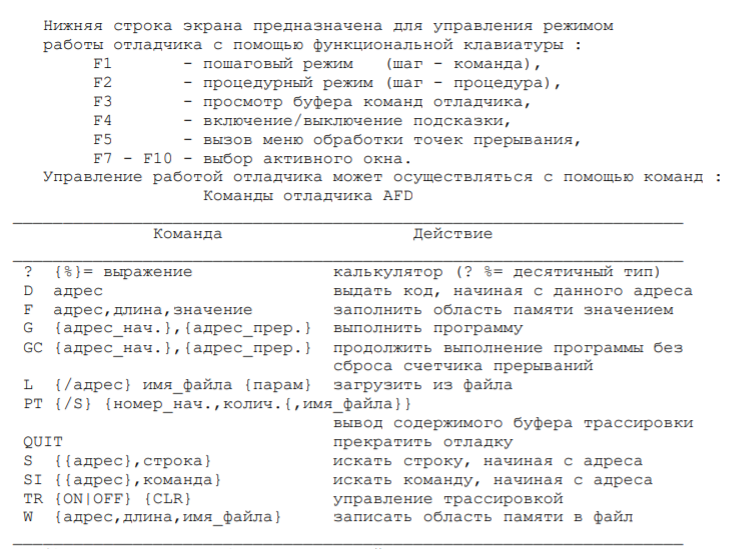
левое - регистры процессора,   
среднее - стек,   
правое - флаги.

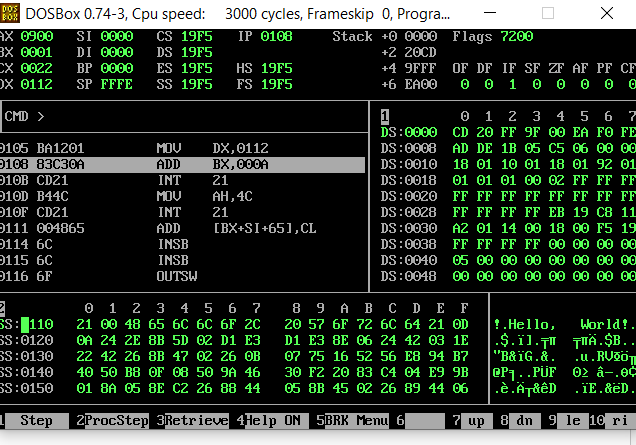
Средний ряд:

левое верхнее - окно команд отладчика,   
левое нижнее - окно кода программы,   
правое - окно данных.

Нижний ряд: окно данных, которое отображает выбранную область памяти в двух форматах: шестнадцатеричном и символьном







$ это конец строки, как и \0 в си

6. Изучите возможности отладчика: шаг с заходом, шаг с обходом, перезапуск программы, создание точек останова, наблюдение за регистрами. Справочная информация по отладчику есть в файле ASM1\_AFD.pdf (выполнять задание из него не нужно).

**Флаг переполнения (Overflow** **Flag или OF)** — указывает на переполнение старшего бита данных (крайнего левого бита) после [**signed**](https://ravesli.com/urok-31-tselochislennyj-tip-dannyh-integer/#toc-2) арифметической операции.

**Флаг направления (Direction** **Flag или DF)** — определяет направление влево или вправо для перемещения или сравнения строковых данных. Если DF = 0, то строковая операция принимает направление слева направо, а когда DF = 1 — строковая операция принимает направление справа налево.

**Флаг прерывания (Interrupt** **Flag или IF)** — определяет, будут ли игнорироваться или обрабатываться внешние прерывания (например, ввод с клавиатуры и т.д.). Он отключает внешнее прерывание, когда значение равно 0, и разрешает прерывание, когда установлено значение 1.

**Флаг ловушка (Trap Flag или TF)** — позволяет настроить работу процессора в одношаговом режиме.

**Флаг знака (Sign** **Flag или SF)** — показывает знак результата арифметической операции. Этот флаг устанавливается в соответствии со знаком элемента данных после выполнения арифметической операции. Знак определяется по старшему левому биту. Положительный результат сбрасывает значение SF до 0, а отрицательный результат устанавливает его равным 1.

**Нулевой флаг (Zero** **Flag или ZF)** — указывает результат арифметической операции или операции сравнения. Ненулевой результат сбрасывает нулевой флаг до 0, а нулевой результат устанавливает его равным 1.

**Вспомогательный флаг переноса (Auxiliary Carry Flag или AF)** — после выполнения арифметической операции содержит перенос с бита 3 на бит 4. Используется для специализированной арифметики. AF устанавливается, когда 1-байтовая арифметическая операция вызывает перенос из бита 3 в бит 4.

**Флаг равенства (Parity** **Flag или PF)** — указывает общее количество 1-бит в результате, полученном после выполнения арифметической операции. Чётное число 1-бит сбрасывает PF до 0, а нечётное число 1-бит устанавливает PF равным 1.

**Флаг переноса (Carry** **Flag или CF)** — после выполнения арифметической операции содержит перенос 0 или 1 из старшего бита (крайнего слева). Кроме того, хранит содержимое последнего бита операции сдвига или поворота.

**Сегмент кода (Code** **Segment или CS)** — содержит все команды и инструкции, которые должны быть выполнены. 16-битный регистр сегмента кода или регистр CS хранит начальный адрес сегмента кода.

**Сегмент данных (Data** **Segment или DS)** — содержит данные, константы и рабочие области. 16-битный регистр сегмента данных или регистр DS хранит начальный адрес сегмента данных.

**Сегмент стека (Stack** **Segment или SS)** — содержит данные и возвращаемые адреса процедур или подпрограмм. Он представлен в виде [**структуры данных «Стек»**](https://ravesli.com/urok-105-stek-i-kucha/). Регистр сегмента стека или регистр SS хранит начальный адрес стека.