

DOSBOX

- `mount c C:\projects\MDL + C:\` – перейти в директорию, указанную в mount.
- `dir` – просмотр списка файлов в директории (аналог ls)

```
Z:\>mount c C:\projects\MDL
Drive C is mounted as local directory C:\projects\MDL\

Z:\>C:\

C:\>dir
Directory of C:\.
.                <DIR>                20-02-2022  15:22
..               <DIR>                20-02-2022  15:20
AFDPRO  EXE             75,894  16-02-2022  10:24
DOSXNT  EXE           393,942  16-02-2022  10:23
HELLO   ASM             154  16-02-2022  10:58
HELLO   COM              28  16-02-2022  11:31
HELLO   OBJ             143  16-02-2022  11:31
LINK    EXE           202,912  16-02-2022  10:24
MASM    EXE            49,152  16-02-2022  10:23
ML      EXE           388,608  16-02-2022  10:23
      8 File(s)          1,110,833 Bytes.
      2 Dir(s)          262,111,744 Bytes free.
```

// При этом эту директорию он будет считать корневой и выйти из нее вниз будет уже нельзя.

- `dir имя файла` – определить размер файла
- Переходы в каталоги
 - `cd путь`
 - `cd\` – переход в корневой каталог
- Создание каталогов
 - `md books` – создать каталог в текущей директории
 - `md путь` – создать в каком-то каталоге
- `rd` – удалить каталог в текущем каталоге
- `cls` – очистить экран
- `copy файл путь+файл` – копирование файлов (есть много ключей)
- `del файл` – удалить файл
- `type файл` – вывести содержимое файла
- `ML.exe /AT main.asm` – скомпилировать программу
- `ML.exe /c main.asm` – ассемблирование программы (трансляция)

```
C:\>ML.exe /c hello2.asm
Microsoft (R) Macro Assembler Version 6.11
Copyright (C) Microsoft Corp 1981-1993. All rights reserved.

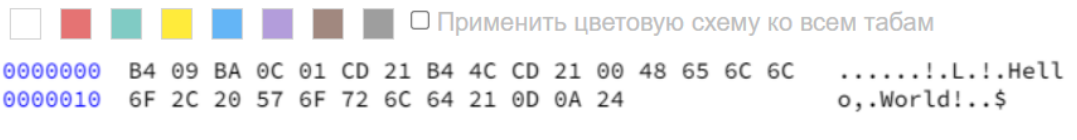
Assembling: hello2.asm
```

- `LINK.exe prog.obj,prog.com,,,nul` – линковка

HEX-VIEWER

<https://hex-works.com/rus>

<https://hexed.it/>



Серенькие стрелные сочетания из двух букв – код символа ASCII в шестнадцатеричном виде (например, код символа W – 57)

Переводчик из текста в коды [здесь](#)

В левой колонке синее – вроде как значение смещения (и одновременно видимо сегменты). Можно заметить, что сереньких циферок в строке всего 16, то есть строки – это сегментные части адреса, столбцы – смещение. В сумме у нас получается адресовать конкретный байт памяти.

Изменять можно значения байт в основной таблице. (Но правда результат вывода пока не очевиден)

Например:

1. Только что скомпилированная программа

| | | | |
|---|----------------------|--------|---------------|
| b4 09 ba 0c 01 cd 21 b4 4c cd 21 00 48 65 6c 6c | ^..í!`lí!·Hell |! | C:\>hello.com |
| 6f 2c 20 57 6f 72 6c 64 21 0d 0a 24 | o, World!..\$. . . . | V..... | Hello, World! |
| | | | C:\>SS |

2. Удалила 6f из второй строчки (из вывода пропала o)

| | | | |
|---|----------------|---------|---------------|
| b4 09 ba 0c 01 cd 21 b4 4c cd 21 00 48 65 6c 6c | ^..í!`lí!·Hell |! | C:\>hello.com |
| 6f 2c 20 57 72 6c 64 21 0d 0a !! | o, Wld!·..... | V·[·..! | Hello, Wld! |
| | | | C:\>S_ |

3. Менять можно не все (такую конструкцию он не дает сохранить)

| | | | |
|---|-----------------|----------|--|
| b4 09 ba 0c 01 00 57 21 b4 4c cd 21 00 48 65 6c | ^..·W!`lí!·Hel | ···%···· | |
| 6c 6f 2c 20 57 72 6c 64 21 0d !! | lo, Wrld!·..... | ····!.. | |

4. Такая попытка провела просто к зависанию программы

| | | |
|----------|---|----------------|
| 00000000 | B4 09 BA 0C 01 B4 21 B4 4C CD 21 00 48 65 6C 6C |!.E!.Hell |
| 0000010 | 6F 2C 20 57 6F 72 6C 64 21 0D 04 34 | .4o,.World!. |

Парсинг программы

Описание основных директив можно найти [здесь](#)

// Указание, что в программе один сегмент на все (стандарт для COM-файла)

.MODEL TINY

// Указание, что сегменты будут сгруппированы в соответствии с соглашениями по упорядочиванию сегментов фирмы Microsoft

.DOSSEG

// Упрощенная директива сегментации (data, code или stack). Не даст создать более трех сегментов в одной программе

.DATA

// msg – просто метка (типа имя переменной)

// db – размер под один символ – 1 байт

// больше всего это похоже на “массив” в памяти из строки и двух регистров, но это только догадка

// см. лекцию 2, псевдокоманды

MSG DB "Hello, World!", 0Dh, 0Ah, '\$'

// Пошел сегмент кода

.CODE

// Обеспечивает код инициализации и отмечает начало программы.

.STARTUP

// положили в регистр ah 09

// 09 – функция вывода строки символов на экран

MOV AH, 09h

// OFFSET – статический способ взять указатель

MOV DX, OFFSET MSG

// Прерывание (в этот момент идет запись в консоль)

INT 21h

// Следующие две строчки нужны для завершения текущего процесса , т.е. прерывания программы

MOV AH, 4Ch

INT 21h

// Этой директивой заканчивается описание любого модуля

END

Отладчик AFDPRO.EXE

TO DO