Работа с видеопамятью

**Видеорежимы**

* Видеоадаптер находится в одном из поддерживаемых им режимов
* Режимы:
  + Текстовые
  + Графические
* Все режимы пронумерованы:
  + Для некоторых режимов характеристики известны заранее
  + Для остальных режимов их можно узнать, взаимодействуя с видеоадаптером

**Видеопамять**

* Видеоадаптер – устройство, преобразующее графический образ в ОЗУ или памяти самого адаптера в форму, пригодную для вывода на экран
  + Современные видеокарты умеют намного больше
* Видеопамять – область памяти, отведенная для хранения данных, которые используются для формирования изображения на экране
  + В первых видеоадаптерах хранилось растровое изображение
  + В современных видеокартах могут храниться также векторные данные (Например, в виде массива треугольников)
* В зависимости от выбранного режима видеоадаптер интерпретирует содержимое видеопамяти по-разному
* Для смены режимов и управления другими параметрами видеоадаптера в реальном режиме используют **int 10h**
  + Соглашение вызова у функций, предоставляемых через **int 10h**, такое же, как у функций MS-DOS
  + Функции **int 10h** – функции BIOS
    - => Функции биос можно использовать даже если нет ОС
  + Для работы с режимами видеоадаптера есть 2 функции:
    - $00 – установить требуемый видеорежим
    - $0F – получить информацию о выбранном сейчас режиме
  + Более полная информация есть в:
    - TechHelp
    - Стандарте VBE
* Текстовый режим 03h
  + По умолчанию в MS-DOS используется видеорежим 03h
    - Текстовый
    - 80x25 знакомест
    - 16 цветов
      * Отдельно задаются цвет символа и цвет фона
    - Видеопамять для этого режима:
      * Начинается с адреса **B800:0000**
      * Является Массивом из пар байтов:
        + Первый байт – код символа
        + Второй байт – цвет

**Еще называют атрибутом**

* + - Запись в видеопамять приводит к изменению изображения на экране

**Работа с видеопамятью**

* Чтение видеопамяти позволяет определить содержимое экрана
  + Но это является нежелательным, особенно в современных видеокартах т.к. может требовать значительных затрат времени
* Для записи в видеопамять удобно использовать строковые команды

**Пример**

* Написать процедуру, которая заполняет последнюю строку экрана символами **=**

**Графический режим 13h**

* Среди стандартных графических режимов самый удобный для использования – режим **13h**
  + Графический
  + 320x200 пикселей
  + 256 цветов
* Видеопамять для этого режима:
  + Начинается с адреса **A000:0000**
  + Является массивом байт
    - Каждый байт – цвет символа

Пример

* Написать программу, которая заполняет экран зеленым цветом

Построение изображений

* Среди функций int 10h есть функции для рисования точек
  + **ОНИ СЛИШКОМ МЕДЛЕННЫЕ!!! НЕ ТРОГАТЬ ИХ!!!**
  + Рекомендуется работать напрямую с видеопамятью
* Удобно реализовывать отдельные процедуры для построения линий, прямоугольников и т.д.
  + **Выносить рисование точки в отдельные процедуры – невыгодно**

**ЧОПОЧИТАТЬ?**

* Алгоритм Брезенхема
  + Используется для рисования отрезков (линий) под произвольным углом
  + Есть также алгоритм Брезенхема для построения окружностей
* 3D-графика
  + Простейший способ перехода от 3D-координат к 2D:
    - [x y z] -> [x/z y/z]
  + Существует много различных видов проекций
    - Общая идея: **вектор** координат точки **умножается на матрицу**, которая задает способ проецирования
* Режимы с высоким разрешением
  + Можно использовать VBE – VESA BIOS Extensions
    - Стандарт есть в общем доступе
  + Для некоторых видеокарт доступна документация
    - **Такое себе**
    - 3Dfx
    - Inel HD Graphics
    - ATI/AMD
    - NVIDIA