ЛАБОРАТОРНАЯ РАБОТА № 12

ПРОГРАММИРОВАНИЕ ГРАФИКИ

Цель работы: Практическое овладение навыками разработки программного кода на языке Ассемблер. Изучение основных принципов представление данных на мониторе компьютера в графическом виде.

Основными задачами выполнения лабораторной работы являются: разработка программы, с использованием графических примитивов.

Результатами работы являются:

- разработанная по индивидуальному варианту программа на языке Ассемблер;
- подготовленный отчет.

Постановка задачи

Построить график функции в соответствии с вариантом, при этом предусмотреть:

a) параметры вводятся с клавиатуры;
b) область построения должна соответствовать построенному графику;
c) наличие осей, дополненных шкалой деления;
d) графики, в разных диапазонах построения должны быть выделены цветом.
e) график строится с эффектом анимации (предусмотреть использование таймера при построении);
f) проверить правильность построения, средствами Excel или MathCad.

КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

EGA

EGA Enhanced Display Adapter — улучшенный графический адаптер выпущен в 1984 году. Он снабжался от 64 до 256 Кб памяти. Позволяет одновременно работать с 26=64 цветами. Яркость изображения на экране определяется уровнем напряжения видеосигнала. Адаптер соединялся с монитором 6 сигнальными проводами (синий, синий вспомогательный, красный, красный

вспомогательный, зеленый и зеленый вспомогательный). Внутри цветного монитора три ЦАПа (цифро-аналоговых преобразователя) позволяли получить из 2-х цифровых сигналов для каждого цвета по 4 (22=4) яркостных градации, которые отправлялись на три цветовые пушки монитора. Адаптер EGA обеспечивает работу монохромного дисплея в графическом режиме, поддерживает все текстовые режимы CGA, графические режимы средней и высокой разрешающей способности CGA (режимы 04 и 05) и к ним добавляются еще 4 собственных графических и 1 текстовый режим.

VGA

Video Graphics Array содержит 256 Кб до 1 Мб памяти. Позволяет выводить на экран 2^{18}=262 144 цветовых оттенка, но одновременно на экране могут быть только 256 цветов. Имеет три встроенных ЦАПа. На монитор VGA адаптер отправляет три аналоговых сигнала, которые управляют работой электронных пушек монитора. Поддерживает 17 документированных режимов: 640x480 (монохром), 640x480x16 цветов, 320x200x256 цветов (таблица 25.) и кучу недокументированных, но также стандартных режимов: 320x400x256 цветов, 360x480x256 цветов и т.д.

Некоторые стандартные видеорежимы VGA и EGA

Название режима	Тип	Разрешающая способность	Количество цветов	Адаптер
07	текстовый	720x350	3 (b/w/bold)	MDA, EGA
0Dh	графический	320x200	16	EGA
0Eh	графический	640x200	16	EGA
0Fh	графический	640x350	3 (b/w/bold)	EGA
10h	графический	640x350	16	EGA
11h	графический	640x480	2	VGA
12h	графический	640x480	16	VGA
13h	графический	320x200	256	VGA

Видеорежимы

Видеорежимы бывают текстовые и графические, различаются разрешением экрана и количеством цветов, одновременно выводимых на экран. В текстовом режиме в видеопамяти находятся коды символов и их атрибуты, которые из таблицы символов выводятся на

экран монитора. В графическом режиме в видеопамяти находится код цвета каждой точки, отображаемой на экране. Видеорежимы меняются из программ с помощью вызова функции *BIOS.*

Значения регистров при вызове функции смены видеорежима:

AH=0 — номер функции *BIOS;*

AL — номер видеорежима, который нужно включить

Фрагмент программы, в котором устанавливается видеорежим 13h.

```
mov AX, 0013h          ; AH=0 AL=13h
int 10h
```

Палитра

VGA адаптер имеет встроенный Цифро-Аналоговый Преобразователь (ЦАП), который содержит 256 регистров цвета. Величина каждого регистра 18 бит. Из этих 18 бит на красный цвет отводится 6 бит, 6 бит — на зеленый цвет и 6 бит — на синий (2^{18}=262144). Номер цвета в видеопамяти — это номер регистра цвета, а цвет точки зависит от значения, хранящегося в этом регистре.

Выбрать палитру — значит установить нужные значения в регистрах цвета. Для установки одного регистра цвета используют *10h* подфункцию *10h* функции *BIOS.*

Значения регистров при вызове функции установки регистра цвета следующие:
AH=10h номер функции;

AL=10h номер подфункции;

BX — номер устанавливаемого регистра;

DH — яркость красного (от *0* до *3Fh*);

CH — яркость зеленого (от *0* до *3Fh*);

CL — яркость синего (от *0* до *3Fh*).

Для установки нескольких или всех регистров цвета используют *12h* подфункцию *10h* функции *BIOS.* Перед вызовом этой функции в памяти создают таблицу палитры, имеющую следующий формат: яркость красного первого устанавливаемого цвета, яркость зеленого первого цвета, яркость синего первого цвета, яркость красного второго цвета и т.д.

Значения таблицы должны находиться в пределах от 0 до 3Fh. Значения регистров при вызове функции установки регистра цвета:
AH=10h номер функции;

AL=12h номер подфункции;

BX — номер первого устанавливаемого регистра;

CX — количество устанавливаемых регистров;

ES:DX — адрес таблицы палитры.

В VGA, даже в текстовых режимах с 16 цветовыми оттенками, вывод цвета осуществляется через ЦАП.

Вывод изображений на экран в графическом режиме
Систему координат экрана в графическом режиме можно представить рис. 1, на котором показан режим 640×350 пикселов. Адресуемым элементом является пиксел, позиция которого определяется номерами столбца и строки.

Рис. 1. Система координат экрана в графическом режиме

Точка

Для построения точки на экран в любом графическом режиме требуется вызвать функцию *0Bh* прерывания *10h*.

Содержание регистров при вызове функции: *AH=0Bh*, в *AL* цвет пикселя (если бит 7=1 выполняется операция логического исключающего ИЛИ с цветом экрана), в *BH* номер страницы, в *CX* — *X* координата пикселя, в *DX* — *Y* координата пикселя. Если Вы хотите писать напрямую в видеопамять, то установка точки в режиме *13h* сводится к записи кода цвета в видеопамять по адресу, соответствующему адресу точки *A=Y*H+X*.

Ниже приводится программа, рисующая три точки разными цветами в разных местах экрана. Подпрограмма Draw_pixel выводит точку с заданными координатами и с заданным цветом. При вызове данной

подпрограммы в регистре BX должна находиться X координата точки, в регистре CL — Y координата, в регистре DL — цвет точки.

Рисование линей на экране

Проведение линий подразумевает установку на экране всех точек, принадлежащих отрезку. Сложность при рисовании линии в том, что точки из которых мы ее строим создавали иллюзию прямой. На экране абсолютно точно можно нарисовать только вертикальные, горизонтальные и 1:1 диагональные линии.

Горизонтальная линия

```
;предварительные установки
PUSH 0A000h
POP ES; позиционируем ES на область видеопамяти
MOV DI,X ; в DI координаты начальной точки по X
MOV AX,320; длина строки экрана
 MUL Y; умножаем на Y
ADD DI,AX; и складываем с X
MOV AL,COLOR; цвет линии
; рисуем горизонтальную линию
MOV CX,N; длина линии
REP STOSB
```

Вертикальная линия

```
MOV CX,N; длина линии
A1: MOV ES:[DI],AL; рисуем точку на строке
ADD DI,320; переход на следующую строку

LOOP A1
```

Диагональная линия с наклоном влево

```
 MOV CX,N; длина линии
A1: MOV ES:[DI],AL; рисуем точку на строке
ADD DI,319; переход на следующую строку
LOOP A1
```

Диагональная линия с наклоном вправо

```
MOV CX,N; длина линии
A1: MOV ES:[DI],AL; рисуем точку на строке
ADD DI,321; переход на следующую строку
LOOP A1
```

ПРАКТИЧЕСКАЯ ЧАСТЬ

Пример: установка регистра цвета (изначально синий) в другое значение (цвет окон изменяется на розовый).

```
 .286
 .model tiny
 .code org 100H
 .startup          ;указывает на начало программы
 mov AX,1010h      ;номер функции
 mov BX,1          ;выбираем первый регистр
 mov DH,3Fh        ;яркость красной компоненты
 mov CH,01Fh       ;яркость зеленой компоненты
 mov CL,01Fh       ;яркость синей компоненты
 int 10h           ;установить нужное значение в
                   ;регистре ЦАП
 RET ;выход из программы
 END
```

Пример: программа, рисующая три точки разными цветами в разных местах экрана. Подпрограмма *Draw_pixel* выводит точку с заданными координатами и с заданным цветом. При вызове данной подпрограммы в регистре *BX* должна находиться *X* координата точки, в регистре *CL* — *Y* координата, в регистре *DL* — цвет точки.

```
.286
.model tiny
.code
 org 100h
.startup
  MOV AH,0Fh       ;запомнить текущий видеорежим
  INT 10h
  MOV VIDEOR,AL    ;видеорежим в переменную videor
  MOV AX, 0013h    ;установить видеорежим 13h
  INT 10h
  PUSH 0A000h      ;установить регистр ES на сегмент
  POP ES           ;видеопамяти
  MOV DI,0         ;установка точки (0, 0) зеленого цвета
  MOV CX,0         ;в левый верхний угол экрана
  MOV DL,2
CALL DRAW_PIXEL
  MOV DI,160       ;установка точки (160,100) красного
  MOV CX,100       ;цвета в центр экрана
  MOV DL,4
  CALL DRAW_PIXEL
  MOV DI,319       ;установка точки (319, 199) белого
  MOV CX,199       ;цвета в правый нижний угол экрана
  MOV DL,7
```

```
CALL DRAW_PIXEL
  XOR AX,AX       ;ожидание нажатия любой клавиши
  INT 16h
  MOV AH,0        ;восстановление видеорежима
  MOV AL,VIDEOR
  INT 10h ret     ;выход из программы
;процедура вывода точки на экран
;DI — X координата точки (от 0 до 319)
;CX — Y координата точки (от 0 до 199)
;DL — цвет точки
PROC NEAR DRAW_PIXEL
;присвоить регистру AX значение горизонтального
разрешения
  MOV AX,320            ;умножение координаты Y
                        ;на горизонтальное 349
  MUL CX                ;результат умножения в AX
  MOV BX,AX             ;сложить с координатой X
  MOV ES:[BX][DI],DL    ;вывести точку на экран
  RET                   ;выход из процедуры
  ENDP                  ;конец процедуры
 VIDEOR DB ?            ;переменная для хранения
                        ;значения текущего видеорежима
END
```

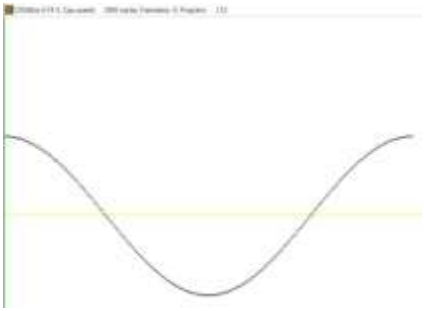
Пример: построения графика функции $F = cos(x)$

```
.486
model use16 small
.stack 100h
.data
b dw 175          ;
k1 dw 1           ;ставим коэффициент K сжатия -
                  ;растяжения по оси Ox
x dw ?
pi dw 180         ;задаём число пи в радианах
y dw ?
axis dw ?         ;задаём ось
k2 dw 70          ;ставим коэффициент K сжатия-
                  ;растяжения по оси Oy
two dw 2
.code
Start:
  mov ax, @data
  mov ds, ax

  xor ax, ax
  mov al, 10h
  int 10h
```

```
  ;Фон
  mov ax, 0600h ; ah = 06 - прокрутка вверх
  mov bh, 15 ;белый
  mov cx, 0000b ; ah = 00 - строка верхнуго левого угла
  mov dx, 184Fh

  int 10h
  mov ah, 0Ch           ;установка графической точки
  mov al, 10            ;загружаем зелёный цвет для
                        ;вертикальной линии
  mov bh, 0h            ;установка номера видеостраницы
  mov cx, 400           ;количество итераций сверху вниз
                        ;для вертикальной линии
@metka1:                ;прорисовка вертикальной линии
  push cx
  mov axis, cx          ;в начало оси записываем 0
  mov dx, axis          ;установка курсора
  mov cx, 0             ;вывод вертикальной оси, со
                        ;сдвигом на 319 вправо
  int 10h
  pop cx                ;400 итераций, ставит в 400
                        ;колонку, и идёт до 0
  loop @metka1
  mov ah, 0ch           ;установка графической точки
  mov al, 30            ;зелёный цвет
  mov cx, 639           ;639 итераций, ставит в 639
                        ;колонку, и идёт до 0
  mov bh, 0h            ;установка номера видеостраницы
  mov dx, 174           ;ставит в 174 строку

@metka2:                ;цикл вывода горизонтальной оси
  int 10h               ;вывод горизонтальной линии
  loop @metka2

  mov cx, 360           ;начинаем рассчитывать функцию
@metka3:                ;отвечает за вывод графа

  mov x, cx       ;помещаем в x 639
  fild x          ;st(0) = 639
  fldpi           ;st(0) = pi, st(1) = 639
  fmul            ;st(0) = 639, st(1) = pi*639
  fild pi         ;st(0) = 180, st(1) = pi*639
  fdiv            ;st(0) = st(1),st(1)=pi*639 / 180
  fild k1         ;st(0) = 2, st(1) = pi*639 / 180
  fdiv            ;st(0) = st(1),
                  ;st(1) =(pi*639)/ (180*2)
  fcos            ;st(0) =cos((pi*639)/(180*2)) = cos(x)
```

```
  fimul k2          ;st(0) = (pi*639*70) / (180*2)
  fchs              ;st(0) = -(pi*639*70) / (180*2)
  fiadd b           ;st(0) = 200 - (pi*639*70) / (180*2)
  frndint[1]        ;округляем st(0) до целого числа
  fistp y           ;st(0) = 0,
                    ;y = 200 - (pi*639*70)/(180*2)
  mov ah, 0Ch       ;установка графической точки
  mov bh, 0h        ;ставим в нулевое окно
  mov dx, y         ;ставим в y строку
  mov al, 0         ;цвет черный
  int 10h

  loop @metka3      ;уменьшаем cx

  mov ah, 8h        ;выход из программы при нажатии любой
                    ;клавиши
  int 21h
mov ax, 4c00h
    int 21h
end Start
end
```

Результат работы программы:



[1] Используйте команды сопроцессора для целых чисел

Задание для самостоятельного выполнения

Вариант 1

$$g=\begin{cases}\dfrac{1+x^2}{\sqrt{1+x^4}}, & x\le 0\\ 2x+\dfrac{\sin^2(x)}{2+x}, & x>0\end{cases}$$

Вариант 2

$$z=\begin{cases}\dfrac{1+|x|}{\sqrt[3]{1+x+x^2}}, & x\le -1,\\ 2\ln\left(1+x^2\right)+\dfrac{1+\cos^4(x)}{2+x}, & x\in(-1,0),\\ (1+x)^{7/5}, & x\ge 0\end{cases}$$

Вариант 3

$$y=\begin{cases}0, & \text{если } x\le -4 \text{ или } x\ge 4\\ -\sqrt{4-(x+2)^2}, & \text{если } -4<x<0\\ \sqrt{4-(x-2)^2}, & \text{если } 0\le x<4\end{cases}$$

Вариант 4

$$z=\begin{cases}\dfrac{1+\sqrt[5]{x}}{3+x^2}, & x<0,\\ \sin^2(x)\sqrt{5+x}, & x\in[0,1),\\ \sin^3(x+1)e^{0,6x}, & x\ge 1.\end{cases}$$

Вариант 5

$$z=\begin{cases}\dfrac{1+x+x^2}{1+x^2}, & x<0,\\ \sqrt{1+\dfrac{5x}{1+x^3}}, & x\in[0,1),\\ 5\left|0,7\cos(x)+\sin(x)\right|, & x\ge 1.\end{cases}$$

Вариант 6

$$z=\begin{cases}3x+\sqrt{1+x^2}, & x<0,\\ 2\cos(x)e^{-2x}, & x\in[0,1],\\ 2\sin(3x), & x>1.\end{cases}$$

Вариант 7

$$z=\begin{cases}\dfrac{|x|}{1+x^2}e^{-5x}, & x<0,\\ \sqrt{1+x^4}, & x\in[0,1),\\ \dfrac{1+\cos(\pi x)}{6+x}+3x, & x\ge 1.\end{cases}$$

Вариант 8

$$z=\begin{cases}\sqrt{1+\dfrac{x^2}{1+x^4}}, & x<0,\\ 2\sin^3(x), & x\in[0,1],\\ \sqrt{1+\left|2\cos(6x)\right|^{1/3}}, & x>1.\end{cases}$$

Вариант 9

$$z=\begin{cases}\sqrt[3]{6+x^2}, & x\le 0,\\ \sin^3(\pi x)+\dfrac{2+x}{1+\cos^2(x)}. \end{cases}$$

Вариант 10

$$z=\begin{cases}\sqrt{1+5x^2-\sin^2(x)}, & x\le 0,\\ \dfrac{(7+x)^2}{\sqrt[3]{4+e^{-0,7x}}}, & x>0. \end{cases}$$

Вариант 11

$$z=\begin{cases}\sqrt{1+x^2}, & x\le 0,\\ \dfrac{1+x^3}{1+\sqrt[5]{1+e^{-0,5x}}}, & x>0. \end{cases}$$

Вариант 12

а

$$z=\begin{cases}1, & \text{если } x^2+y^2\le 1\\ x^2+y^2, & \text{если } 1<x^2+y^2<4\\ 4, & \text{если } x^2+y^2\ge 4 \end{cases}$$

Вариант 13

$$y=\begin{cases}1/(x-2)^2 & \text{если } x<0 \text{ или } x\ge 4\\ x^2+4x-7, & \text{если } 0<x<2\\ 1/(x^2+4x-7), & \text{если } 2\le x<4 \end{cases}$$