

Университет ИТМО
Факультет Компьютерных Технологий и Управления
Кафедра Вычислительной техники

Лабораторная работа №1
по дисциплине «Компьютерная графика»

Студент: Широков О.И.
Группа: р3411

Санкт-Петербург
2015г.

Оглавление

1.	Задание	2
2.	Выполнение	2
3.	Вывод	7

1. Задание

- 1 В видеоадаптере CGA (черно-белом) в видеопамяти на фоне всех нулей в байт с адресом $2000h+83h$ относительно начала видеопамяти записан код $19h$. Определить, что отобразится на экране в этом случае, и обоснования этого определения представить преподавателю.
- 2 Разработать программу для этого же видеоадаптера для вывода на экран точки с координатами $x = 204$, $y = 111$.
- 3 На базе разработанной программы вывода точки разработать программу построения горизонтальной линии произвольного размера от минимального до максимального.
- 4 На базе разработанной программы вывода точки разработать программу построения вертикальной линии произвольного размера от минимального до максимального.
- 5 Получить у преподавателя задание на построения фигуры на базе процедур построения горизонтальной и вертикальной линий.

2. Выполнение

```
1 .286
2 .model small
3 .code
4     mov ax, 0006h
5     int 10h
6
7     mov ax, 0b800h
8     mov es, ax
9
10    mov di, 2083h
11    mov al, 19h
12
13    stosb
14
15    xor ax, ax
16    int 16h
17
18    mov ax, 0003h
19    int 10h
20
21    mov ax, 4c00h
22    int 21h
23 end
```

Листинг 1.1: Программа для отображения заданной точки на экране

```
1 .286
2 .model small
3 .code
4     mov ax, 0006h
5     int 10h
6
7     mov ax, 0b800h
8     mov es, ax
9
10    mov di, 3149h ; (204, 111)
11    mov al, 40h
12
13    stosb
```

```

14
15     xor ax, ax
16     int 16h
17
18     mov ax, 0003h
19     int 10h
20
21     mov ax, 4c00h
22     int 21h
23 end

```

Листинг 1.2: Программа для отображения точки на экране

```

1 .286
2 .model small
3
4 .code
5     mov ax, 0006h
6     int 10h
7
8     mov ax, 0b800h
9     mov es, ax
10
11     ; call draw_horizontal_line
12     call draw_vertical_line
13
14     xor ax, ax
15     int 16h
16
17     mov ax, 0003h
18     int 10h
19
20     mov ax, 4c00h
21     int 21h
22
23 draw_horizontal_line:
24     push bp
25     mov bp, sp
26     sub sp, 7
27
28     mov word ptr [bp - 2], 05h; X start coord
29     mov word ptr [bp - 4], 7Fh; Length of line
30     mov byte ptr [bp - 5], 5Fh; Y start coord
31     mov byte ptr [bp - 6], 00h; value of first byte
32     mov byte ptr [bp - 7], 00h; value of last byte
33
34     ; Cut length
35
36     mov ax, word ptr [bp - 4]
37     mov bx, word ptr [bp - 2]
38
39     add ax, bx
40     cmp ax, 280h
41     jl less_than_width
42
43     mov ax, 280h
44     sub ax, bx
45
46     mov word ptr [bp - 4], ax
47
48     less_than_width:
49
50     ; Compute last byte
51

```

```

52     add ax, bx
53     and al, 07h
54
55     mov cl, al
56
57     mov al, 0FFh
58     shr al, cl
59     not al
60
61     mov byte ptr [bp - 7], al
62
63     ;Compute first byte
64
65     mov cx, word ptr [bp - 2]
66     and cl, 07h
67
68     mov al, 0FFh
69     shr al, cl
70
71     mov byte ptr [bp - 6], al
72
73     ;Compute address
74
75     mov al, byte ptr [bp - 5]
76     shr al, 1
77
78     mov bl, 50h
79     mul bl
80
81     mov di, ax
82
83     mov al, byte ptr [bp - 5]
84     and al, 01h
85     jz even_line
86
87     add di, 02000h
88
89     even_line:
90
91     mov al, byte ptr [bp - 2]
92     shr al, 3
93
94     xor ah, ah
95     add di, ax
96
97     ;Drawing!
98
99     mov al, byte ptr [bp - 6]
100
101     mov bl, byte ptr [bp - 4]
102     cmp bl, 01h
103     je draw_one_byte
104     jl drawing_end
105     jne draw_some_bytes
106
107     draw_one_byte:
108
109     mov bl, byte ptr [bp - 7]
110     and al, bl
111
112     stosb
113     jmp drawing_end
114

```

```

115     draw_some_bytes:
116
117     mov bx, word ptr [bp - 4]
118
119     drawing_loop:
120         stosb
121
122         mov al, 0FFh
123         sub bx, 8
124         cmp bx, 0
125         jg drawing_loop
126
127     mov al, byte ptr [bp - 7]
128     stosb
129
130     drawing_end:
131     mov sp, bp
132     pop bp
133     ret
134
135 draw_vertical_line:
136     push bp
137     mov bp, sp
138     sub sp, 5
139
140     mov word ptr [bp - 2], 0FFh; X
141     mov byte ptr [bp - 3], 00h; Y
142     mov byte ptr [bp - 4], 8Fh; Len
143     mov byte ptr [bp - 5], 00h; byte value
144     mov word ptr [bp - 7], 00h; base address
145
146     ;Byte value
147
148     mov al, 80h
149     mov bx, word ptr [bp - 2]
150
151     mov cl, bl
152     and cl, 07h
153
154     shr al, cl
155     mov byte ptr [bp - 5], al
156
157     ;Byte address, X is in bx
158
159     shr bx, 3
160
161     mov al, byte ptr [bp - 3]
162     shr al, 1
163
164     mov cl, 50h
165
166     mul cl
167
168     add bx, ax
169
170     mov word ptr [bp - 7], bx
171
172     ;Cut len
173
174     ;mov al, byte ptr [bp - 3]
175     ;mov ah, byte ptr [bp - 4]
176     ;add ah, al
177

```

```

178     xor ax,ax
179     xor bx,bx
180
181     mov al, byte ptr [bp - 3]
182     mov bl, byte ptr [bp - 4]
183
184     add bx, ax
185
186     cmp bx, 0C8h
187     jle less_than_height
188
189     xor ax, ax
190     xor bx, bx
191
192     mov al, byte ptr [bp - 3]
193     mov bl, 0C8h
194
195     sub bx, ax
196
197     mov byte ptr [bp - 4], bl
198
199     less_than_height:
200
201     xor ax, ax
202     mov al, byte ptr [bp - 5]
203
204     xor cx, cx
205
206     mov ch, 02h
207
208     draw_loop:
209
210         mov cl, byte ptr [bp - 4]
211         shr cl, 1
212
213         mov di, word ptr [bp - 7]
214
215         mov dl, ch
216         and dl, 01h
217         jz draw_half
218
219         add di, 2000h
220
221         draw_half:
222
223             mov es:[di], al
224             add di, 50h
225
226             dec cl
227             jnz draw_half
228
229         dec ch
230         jnz draw_loop
231
232     mov sp, bp
233     pop bp
234     ret
235 end

```

Листинг 1.3: Программа для отображения заданной точки на экране

3. Вывод

В ходе выполнения ЛР были исследованы особенности работы с СГА.