

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

**ЛАБОРАТОРНАЯ РАБОТА №5 ПО ПРЕДМЕТУ
«МАШИННО-ЗАВИСИМЫЕ ЯЗЫКИ ПРОГРАММИРОВАНИЯ»**

студента 2 курса 251 группы
направления 09.03.04 — Программная инженерия
факультета КНиИТ
Ланкина Ивана Сергеевича

Проверила

Т. С. Бибичева

1 Задание

Вариант 3. Изображение показано на рисунке 5.6. и состоит из 10 строк символов начиная с символа D (ASCII 44h) и далее по алфавиту с разными атрибутами начиная с 05h и далее плюс один. В каждой строке по 10 символов, начальная позиция вывода 10:20. Надо выполнить задание, используя прямую работу с видеопамятью (см. пример 5.2)

2 Алгоритм

1. Определяем текущий видеорежим и текущую активную страницу
Сохраняем в стеке
Ставим стандартный цветной текстовый режим и первую страницу
2. Запускаем процедуру, в которой происходит следующее:
Помещаем в стек все регистры общего назначения
Помещаем символ D в регистр AL
Помещаем значение атрибута в регистр AL
Устанавливаем начальное смещение и значение CX
Начинаем вывод всего текста построчно, увеличивая CX по 1 до 10
В этом цикле делаем цикл для вывода строки посимвольно, помещая AL
в 1й байт, AH - во 2й байт и смещаясь к след позиции
После каждой строки переходим к след. символу и атрибуту
Если мы не дошли до последней 10й строки переходим в начало цикла
Иначе восстанавливаем все регистры общего назначения и выходим из
процедуры
3. Устанавливаем режим ожидания нажатия клавиши
После нажатия восстанавливаем начальный видеорежим
Завершаем программу

3 Текст программы

```
1  .model small
2
3  .stack 100h
4  .186
5
6  .code
7
8  start:
9  mov AX, @data                ;который сначала загрузим в AX,
10 mov DS,AX                   ;и переложим в DS
11 mov AX,0b900h               ;Используя сегментный регистр ES,
12 mov ES,AX                   ;организовать запись данных в видеопамять
13 mov AH, 0Fh                 ;определяем текущий видеорежим и текущую активную
    ↪    страницу
14 int 10h
15 push AX                     ;сохраняем в стеке
16 mov AH, 00h                 ;Запрос на установку видеорежима
17 mov AL, 03                   ;Стандартный цветной текстовый режим
18 int 10h
19 mov AH,05h                  ;Выбор функции для вывода страницы
20 mov AL,01h                  ;Страница 1
21 int 10h                     ;Вызвать обработчик прерывания
22 call B10DISPLAY
23 mov AH, 10h                 ;ожидание нажатия клавиши
24 int 16h
25 pop AX
26 mov AH, 00h                 ;восстанавливаем видеорежим
27 int 10h
28
29 mov ax,4C00h
30 int 21h
31
32
33 B10DISPLAY proc              ;начало процедуры - Обработка области
    ↪    вывода и вывод
34     pusha                    ;помещаем в стек
        ↪    все регистры общего назначения
35     mov AL, 44h              ;помещаем символ D
        ↪    в регистр AL
```

```

36      mov AH, 05h                                ; помещаем значение
        ↳ атрибута в регистр AL
37      mov DI, 1640                                ; начальное смещение
38      mov CX, 0                                    ; инициализируем CX
        ↳ значением 0
39      pr:                                          ; цикл для вывода всего
40      push CX                                      ; сохраняем
        ↳ значение CX в стеке
41      mov CX, 10                                  ; меняем на число
        ↳ символов в строке
42      str_out:                                    ; цикл для вывода строки
43      mov ES:word ptr[DI], AX                    ; AL поместить в 1й байт, AH
        ↳ поместить во 2й байт
44      add DI, 2                                    ; переход к след.
        ↳ позиции в строке
45      loop str_out                                ; пока CX не 0 переход к
        ↳ началу цикла и уменьшить CX на 1
46      add DI, 140                                  ; сдвиг до начала
        ↳ след строки
47      inc AL                                       ; увеличиваем
        ↳ символ до след.
48      inc AH                                       ; увеличиваем
        ↳ атрибут до след.
49      pop CX                                       ; восстанавливаем
        ↳ CX (номер строки)
50      inc CX                                       ; увеличиваем на
        ↳ 1
51      cmp CX, 10                                  ; если это не 10я
        ↳ строка (сравниваем CX с 10)
52      jne pr                                       ; переходим в
        ↳ начало цикла
53      popa                                        ; восстанавливаем
        ↳ все регистры общего назначения
54      ret                                         ; возврат из
        ↳ подпрограммы в точку вызова
55 B10DISPLAY endp                                ; конец процедуры
56
57 end start

```

4 Пример работы

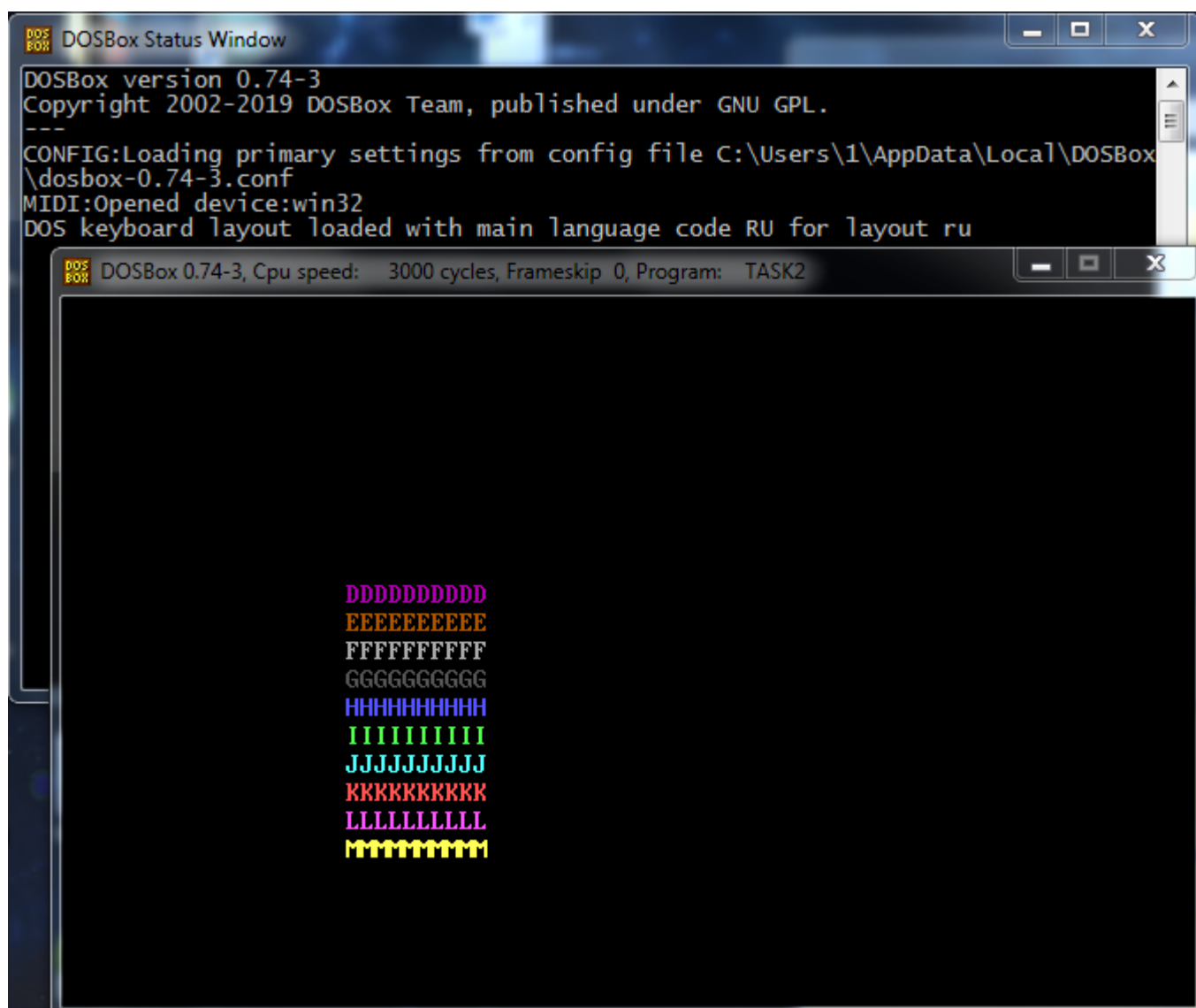


Рисунок 1 – Пример работы программы

5 Контрольные вопросы

1. Каков адрес области видеоданных для режимов 00h - 06h — B800h, монохромного текстового режима — B000h;
2. 4 страницы, разрешение 720*400, число цветов 16
3. 00110101
11110010
4. Бит 7 (BL) устанавливает атрибут мерцания (может быть заблокирован). Биты 6 – 4 определяют цвет фона символа.
Бит 3 (I) устанавливает для символа нормальную (0) или повышенную (1) яркость.
Биты 2 – 0 определяют составляющие цвета символа.
5. 10 желтых сердечек (03h) на синем фоне:
mov AH, 09h ;Запросить вывод
mov AL, 03h ;Выводимый символ
mov BH, 0
mov BL, 1Eh ;устанавливаем цвет фона и символов
mov CX, 10 ;10 символов
int 10h
10 белых звездочек (2Ah) на красном фоне
mov AH, 09h ;Запросить вывод
mov AL, 2Ah ;Выводимый символ
mov BH, 0
mov BL, 47h ;устанавливаем цвет фона и символов
mov CX, 10 ;10 символов
int 10h