

Министерство образования и науки РФ
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования

Тульский государственный университет

КАФЕДРА ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

МЕТОДЫ ПРОЕКТИРОВАНИЯ БЕЗОПАСНЫХ TSR-ПРОГРАММ ДЛЯ DOS

Лабораторная работа № 8
по курсу «Методы и средства защиты компьютерной информации»

Выполнил:	студент группы 220601	_____	Белым А.А.
		(подпись)	
Проверил:	д. т. н., проф. каф. ВТ	_____	Данилкин Ф.А.
		(подпись)	

Тула 2013

Цель работы

Получение навыков разработки и отладки TSR-программ для DOS.

Задание

Разработать резидентную программу, обеспечивающую регулировку контрастности изображения для видеорежимов 10h, 12h.

Текст программы

Далее представлен текст резидентной программы на языке C++, обеспечивающей регулировку контрастности изображения.

```
#include <stdio.h>
#include <mem.h>
#include <dos.h>
#define regs void
#define CNUM 6
#define TSR_MARK 0xDEAD
unsigned char far *key_stat=(unsigned char far*)0x00400017L;
unsigned int far *TSR_maddr=(unsigned int far*)0x0040005CL;
void interrupt (*oldvec9)(...),(*oldvec8)(...);
int exiting;
unsigned PSP;
int index=0;
void readDAC(unsigned char *p,unsigned start,unsigned count){
    asm{
        mov ax,SS
        mov ES,ax
        mov dx,p
        mov ax,1017h
        mov bx,start
        mov cx,count
        int 10h
    }
}

void writeDAC(unsigned char *p,unsigned start,unsigned count){
    asm{
        mov ax,SS
        mov ES,ax
        mov dx,p
        mov ax,1012h
        mov bx,start
        mov cx,count
        int 10h
    }
}

void selectDACpagemode(unsigned char pmode){
    asm {
        mov ax,1013h
        mov bh,pmode
        mov bl,00h
        int 10h
    }
}

#define DAC_PAGE(i) ((i)>0?(i)+3:((i)?16+(i):0))
void selectDACpage(unsigned char page){
```

```

    asm {
        mov ax,1013h
        mov bh,page
        mov bl,01h
        int 10h
    }
}
void contrastmod(float c,int index){
    if(!index)
        return;
    unsigned char DAC[3*16];
    for(int j=0;j<3*16;j++){
        DAC[j]=0;
    }
    readDAC(DAC,0,8);
    readDAC(DAC+3*8,56,8);
    float f;
    for(int i=0;i<3*16;i++){
        f=(DAC[i]-31.5)*c+31;
        DAC[i]=f>=63?63:(f<=0?0:f);
    }
    writeDAC(DAC,DAC_PAGE(index)*16,16);
}
void selectPalette(int index){
    if(index){
        selectDACpagemode(1);
        selectDACpage(DAC_PAGE(index));
    } else {
        selectDACpagemode(0);
        selectDACpage(0);
    }
}
void increaseContrast(int& index){
    if(index<CNUM){
        selectPalette(++index);
    }
}
void decreaseContrast(int& index){
    if(index>=CNUM){
        selectPalette(--index);
    }
}
}

void interrupt vec8(...){
    setvect(0x8,oldvec8);
    oldvec8(...);
    if(exiting){
        selectPalette(0);
        setvect(0x9,oldvec9);
        *TSR_maddr=0;
        //outp(0x20,0x20);
        asm{
            mov es,PSP
            mov es,es:[2Ch]
            mov ah,49h
            int 21h

            mov es,PSP
            mov ah,49h
            int 21h

            //mov ax,4c00h
            //int 21h
        }
    }
}
}

```

```

void setTimer(){
    if(getvect(0x8)!=vec8){
        oldvec8=getvect(0x8);
        setvect(0x8,vec8);
    }
}
void interrupt vec9(...){
    //unsigned ax=reg.x.ax
    unsigned char c=inp(0x60);
    if((*key_stat)&4){
        switch(c){
            case 61:
                decreaseContrast(index);
                outp(0x20,0x20);
                return;
            case 62:
                increaseContrast(index);
                outp(0x20,0x20);
                return;
            case 63:
                exiting=1;
                setTimer();
                outp(0x20,0x20);
                return;
        }
    }
    oldvec9();
}

const float mod_n[CNUM]={0.9,0.75,0.5,0.25,0.1,0.0},
            mod_p[CNUM]={1.25,1.5,2.0,4.0,8.0,16.0};
int main(){
    PSP=_psp;
    if(*TSR_maddr!=TSR_MARK){
        printf("Installing as resident...\n");
        *TSR_maddr=TSR_MARK;
        for(int i=0;i<CNUM;i++){
            contrastmod(mod_p[i],i+1);
            contrastmod(mod_n[i],-i-1);
        }
        exiting=0;
        oldvec9=getvect(0x9);
        setvect(0x9,&vec9);
        keep(0,_SS+_SP/16-_psp);
    } else
        printf("Resident already in memory!\n");

    return 0;
}

```

Тестовый пример

На рисунке 1 представлен пример работы программы при снижении контрастности изображения.

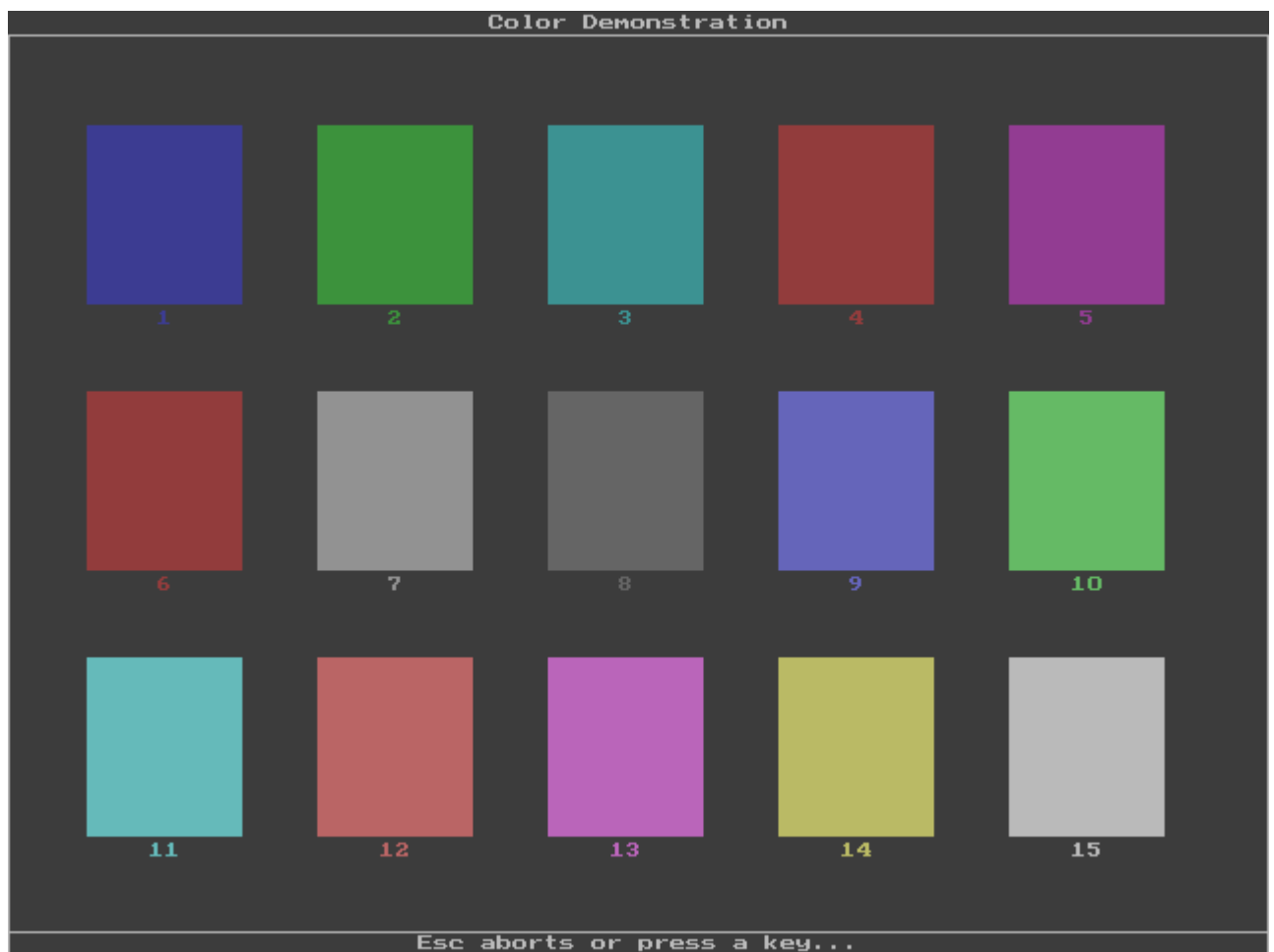


Рисунок 1 – Пример изображения пониженной контрастности

На рисунке 2 представлен пример работы программы при повышении контрастности изображения.

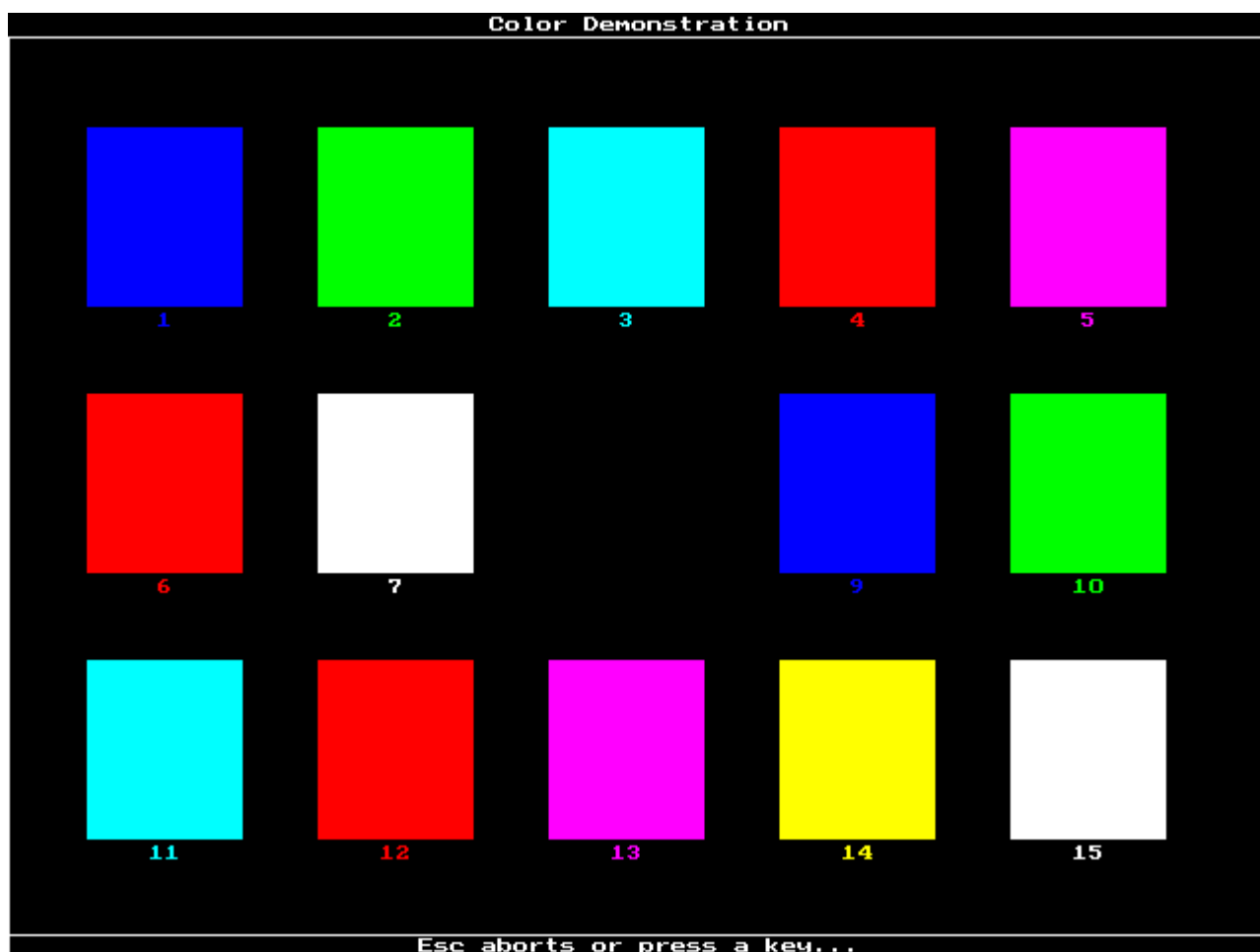


Рисунок 2 – Пример изображения повышенной контрастности

Вывод

В данной работе рассмотрены принципы и приемы создания резидентных программ для DOS. Резидентные программы позволяют выполнять фоновые задачи в однозадачной операционной системы, и поэтому могут быть использованы для скрытых вредоносных действий. В данной работе разработана резидентная программа, регулирующая контрастность изображения.