

1) Se pretende desarrollar un programa que simplemente aguarde la opresión de teclas hasta que detecte ESCAPE (decimal 27) para finalizar. Además, mientras se reciben las teclas deberá:

- Imprimir un mensaje, SI CORRESPONDIERE, sobre stdout **CADA 10 SEGUNDOS** que informe: **“NO SE HA OPRIMIDO UNA TECLA”**. El mensaje deberá aparecer a partir de la fila 5 columna 10, y deberá cambiar de color ciclicamente por cada aparición.
- Generar una señal cuadrada sobre el **b0 de BASE del LPT (0x378)**.de periodo T=10 seg donde se encuentra conectado un zumbador. Los primeros 5 segundos la amplitud será cero. Repetir ciclicamente mientras no se oprima tecla. Debe tener en cuenta que **NO DEBE** alterar el resto de los bits de BASE.

**NOTA:** Sin importar la cantidad de segundos transcurridos, al llegar tecla analizarla y reiniciar la temporización si correspondiere.

Para realizarlo, se dispone del siguiente esqueleto de programa **INMODIFICABLE**, y que Ud. deberá **COMPLETAR** para que funcione según lo solicitado.

```
#include <dos.h>
#define IRQ8 0

void Print (char * cadena, char f, char c , char atributo); // escribe en la memoria de video
void Inicializar (void); //inicializa el hardware convenientemente
void Salir (void); //deja el hardware como estaba
void interrupt far Newtick (void); //función de atención de interrupción
.....
```

```
int ticks= ???;
char color=0;

void main (void)
{
    char salida = 1;
    Inicializar ( );
    while (salida)
    {
        if (kbhit ())
        {
            .....
            .....
            salida=0;
            ticks=???;
        }
    }
    Salir( );
}
```

Se prohíbe el uso de gettime(), delay() y sleep()

*“El sistema debe quedar igual que antes de comenzar”*

### MEMORIA DE VIDEO

Dirección de comienzo: 0xB8000000L

Tamaño: 4Kb

Disposición: carácter – atributo

Configuración del atributo:

b7	b6	b5	b4	b3	b2	b1	b0
B	FONDO			I	FRENTE		

**B= parpadeo**  
**I = Intensidad**

2)

1. Genere un grupo de funciones que permitan la construcción de una *PILA*: putfirst(), dellist(), printlist(); considerando que el **NODO** tiene el siguiente formato:

```
struct nodo
{
    DATO d;
    struct nodo *next;
};
```

```
typedef struct dato
{
    int i;
    flota f;
    char A[10];
}DATO;
```

**NOTA** Las funciones **NO!!!!!!** pueden usar variables globales.

2. Utilizando las funciones desarrolladas en el punto 1, genere un programa que ingrese desde la línea de comandos un nombre de archivo y permita guardar los datos en orden inverso en otro archivo con extensión **xxx**.

**NOTA(i):** El formato del archivo se corresponde con el apropiado para guardar estructuras del tipo DATO del punto 1.

```

#include<dos.h>
#include<conio.h>
#define IRQ0      8
void Print (char*,char,char,char);
void (interrupt far *Oldtick)(void);
void interrupt far Newtick (void);
void Inicializar (void);
void Quit (void);
int ticks=182;
char color=0;
outportb(0x378,0); //inicializo.
void main (void)
{
    char salida = 1;
    Inicializar ();
    clrscr();
    while (salida)
    {
        if(kbhit())
        {
            tecla=getch();
            if (tecla == 27)
                salida=0;
            ticks=182;
            outportb(0x378,
inportb(0x378)& 0xfe );
        }
        Salir ();
    }
void Inicializar (void)
{
    disable();
    Oldtick=getvect(IRQ0);
    setvect (IRQ0, Newtick);
    enable();
}
void Salir (void)
{
    disable();
    setvect (IRQ0, Oldtick);
    enable();
}

```

```

void Print (char* msg, char f, char c, char color)
{
    char far *p=0xB8000000L;
    p=p+2*c+f*160;
    while (*msg!= '\0')
    {
        *p=*msg;
        p++;
        *p=color;
        p++;
        msg++;
    }
}
void interrupt far Newtick (void)
{
    ticks--;
    if(ticks==91)
        outportb(0x378, inportb(0x378)|0x01);
    if(!ticks)
    {
        Print("NO HA OPRIMIDO TECLA",5,10,color);
        color++;
        if(color==7)
            color=0;
        outportb(0x378, inportb(0x378)& 0xfe );
        ticks=182;
    }
    Oldtick();
}

```