

Tarea 1 ILI-253

Lenguajes de Programación

Primer semestre, 2014

Francisco Bórquez Álvaro Hernández
francisco.borquez@usm.cl alvaro.hernandez@alumnos.usm.cl

Teodoro Saavedra
teodoro.saavedra@alumnos.usm.cl

25 de marzo de 2014

1. Objetivos

- *Recordar y reforzar los conocimientos del lenguaje de programación C.*
- *Aprender el uso de librerías externas para la construcción de software.*
- *Conocer el uso de punteros a funciones.*

2. Tarea

Los alumnos deberán realizar un programa el cual muestre paletas de colores, ordenadas tanto como en una lista como en un vista grid, utilizando la librería ncurses.

3. Desarrollo

El programa consistirá en dos partes, la primera parte será en usar parámetros por línea de comandos (argumentos de la función main) para insertar o eliminar colores.

```
-i <nombre_color><num_red><num_green><num_blue>
```

Este comando permitirá ingresar un color a un archivo simple de texto llamado `colors.txt` que contendrá todos los colores ingresados y donde `<nombre_color>` es el nombre que se le colocara al color, y `<num_red>`, `<num_green>`, `<num_blue>` son números de 0 a 1000 (*ver documentación de ncurses, especialmente la función `init_color`*). Y el formato del archivo será:

```
<nombre_color1>  
<num_red>  
<num_green>  
<num_blue>
```

Ejemplo :

```
-i rojo 1000 0 0
```

Ingresa el color rojo al archivo

```
Azul  
0  
0
```

1000

Rojo

1000

0

0

Por otro lado se podrá eliminar colores del archivo con el comando:

`-d <nombre_color>`

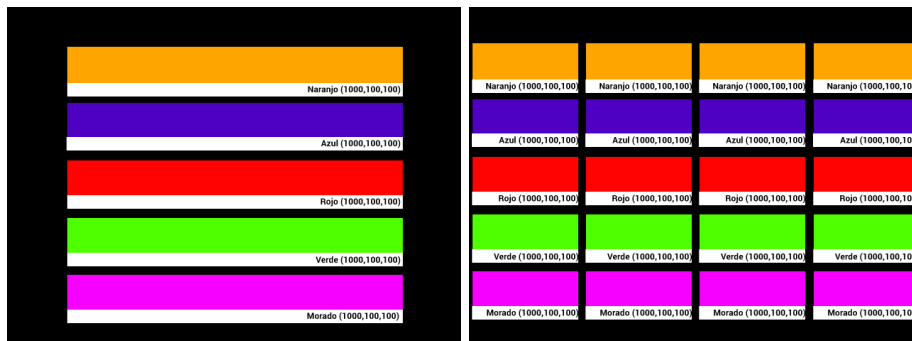
Lo que convierte al `<nombre_color>` en un identificador único. Para hacer tanto la inserción como la eliminación de colores deberán crear una lista enlazada simple de la estructura color para luego realizar dicha acción y volver a escribir el resultado al archivo `colors.txt`.

```
Struct color{
char* nombre;
int red;
int green
int blue;
Color (*newColor) ()
}
```

La segunda parte del programa debe ser llamada por los comandos `-grid` o `-list`, lo que llamara al programa a mostrar los colores ingresados en una lista o en un grid, para esto se deberá crear una estructura encargada de todo el procedimiento de impresión de dichas vistas, la cual se llamara dibujante:

```
Struct dibujante{
Int params
void (*DrawGrid) (void*)
void (*DrawList) (void*)
}
```

la vista grid deberá contener 4 columnas aprox. y 4 filas, por otro lado la lista solo 5 filas, si es que la cantidad de colores no alcanza el programa deberá generar más paginas, las cuales se podrá navegar por medio de botones definidos pudiendo ir tanto hacia delante como hacia atrás. Por último cada color deberá tener su nombre y los números de los colores asignados a él en dichas vistas. El ejemplo de las figuras muestra lo que se busca implementar, teniendo a la izquierda la lista y a la derecha la vista grid:



4. Requerimientos adicionales

- Se debe implementar una lista doblemente enlazada para la manipulación de los colores. Esta debe ser de largo variable, por lo cual se requiere el uso de memoria dinámica.

- No está permitido el uso de librerías externas para el manejo de dicha lista.
- El programa debe incluir un MAKEFILE para su compilado.
- Como la lista enlazada no es parte del programa en sí, debe estar en archivos separados, con su respectiva separación en .h y .c.
- Cada definición de estructuras y cabeceras de funciones deben estar en un archivo .h.
- Se debe liberar la memoria al momento de finalizar el programa.

5. Archivos a entregar

- Color.c
- Dibujador.c
- Main.c
- List.c
- Incluyendo los archivos de cabecera .h para Color.c, Dibujador.c y List.c.
- MAKEFILE Archivo para la compilación del programa
- README.txt.

6. Sobre la entrega

- La revisión se realizara usando como sistema operativo el disponible en los laboratorios del campus.
- El código debe venir indentado y sin warnings.
- Cada función debe llevar una descripción según lo siguiente:

```
/****** Funcion: Nombre_Funcion *****/
Descripcion: Funcion en cargada de...
Parametros:
n1 entero
n2 entero
Retorno: Retorna...
*****/
```

- Debe estar presente y **funcionando** el archivo MAKEFILE para que se efectuó la revisión.
- Se debe trabajar en grupos de dos personas.
- La entrega debe realizarse en un archivo **.tar.gz** y debe llevar el nombre **Tarea1_Rol-1_Rol-2**.
- El archivo README.txt debe contener el nombre y rol de los integrantes del grupo, como también las teclas asignadas a cada funcionalidad del reproductor.
- La entrega se debe entregar antes de las **23.55 hrs.** del día **Martes 8 de Abril del 2014**.
- Por cada día de atraso se descontaran 20 puntos.
- Las copias serán evaluadas con nota 0.