

Programación en C++

Ncourses

Universidad Nacional de las Artes (UNA)

Artes Multimediales

Informática General

Profesor Juan José María Tirigall

Ncurses

Ncurses es una biblioteca de programación que provee una API que permite al programador escribir interfaces basadas en texto. También optimiza el refresco de la pantalla, lo que permite reducir la latencia experimentada. Ncurses significa «new curses», ya que es un reemplazo del discontinuado curses. Ncurses es un clon del original 'System V Release 4.0 (SVr4)' Curses, el cual fue una mejora sobre el clásico discontinuado '4.4 BSD Curses'. Resumiendo, es una librería de funciones que maneja la pantalla de una aplicación para terminales basadas en texto, que incluye posicionamiento del cursor, manejo de colores, refresco de pantalla, detección de teclas sin tener que presionar la tecla *enter*, etc.

Instalación de la librería Ncurses

Ejecutar en la terminal de Linux:

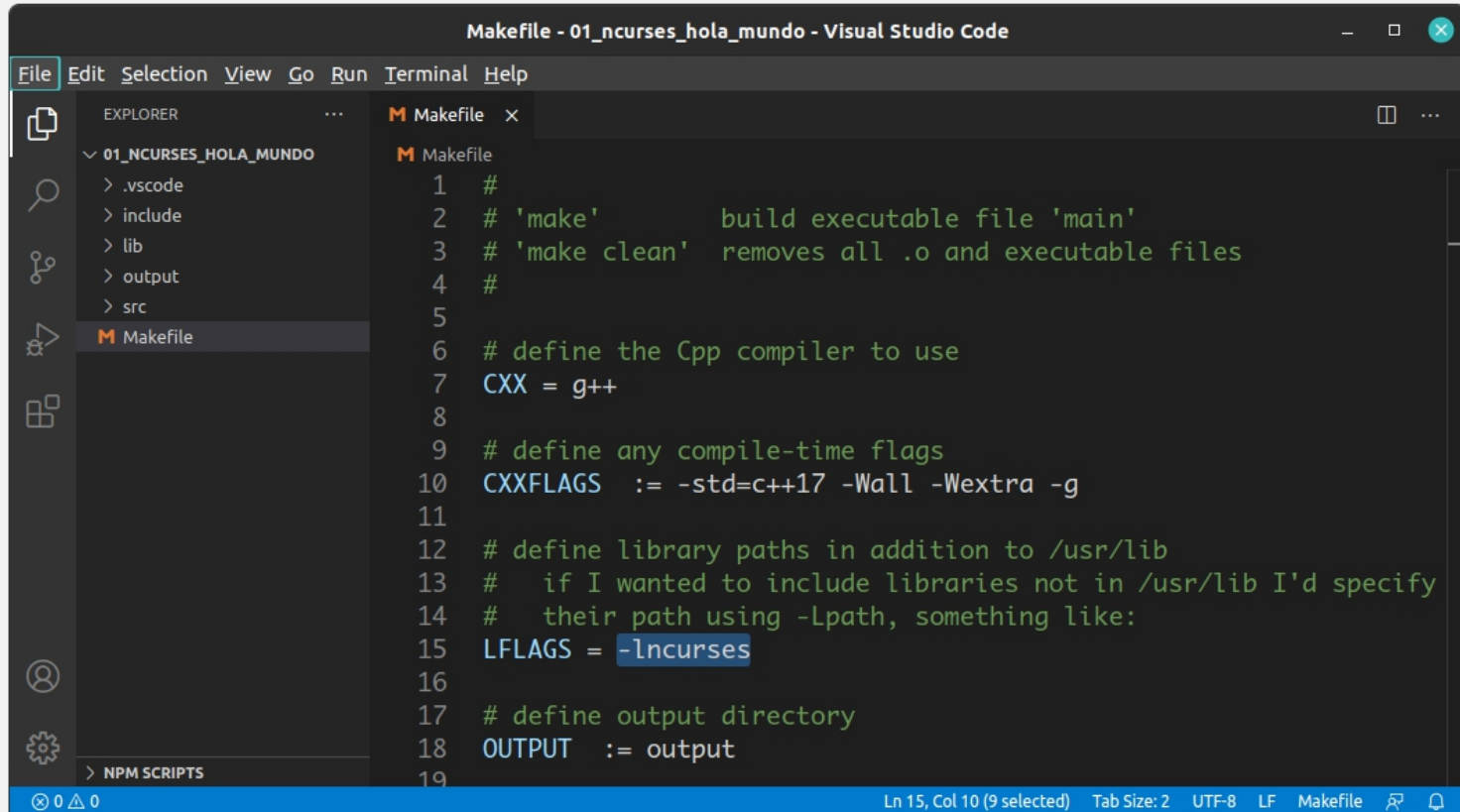
```
sudo apt-get update
```

```
sudo apt-get upgrade
```

```
sudo apt-get install libncurses5-dev libncursesw5-dev
```

Instalación de la librería Ncurses

Creamos un proyecto de C++ nuevo con VSCode y modificamos el archivo Makefile donde dice: **LFLAGS = -lncurses**



The screenshot shows the Visual Studio Code interface with a project named '01_ncurses_hola_mundo'. The Explorer sidebar on the left shows the project structure with folders like .vscode, include, lib, output, and src, and a file named Makefile. The main editor window displays the content of the Makefile. The line 'LFLAGS = -lncurses' is highlighted in blue. The status bar at the bottom indicates 'Ln 15, Col 10 (9 selected)'.

```
1 #
2 # 'make'      build executable file 'main'
3 # 'make clean' removes all .o and executable files
4 #
5
6 # define the Cpp compiler to use
7 CXX = g++
8
9 # define any compile-time flags
10 CXXFLAGS := -std=c++17 -Wall -Wextra -g
11
12 # define library paths in addition to /usr/lib
13 #   if I wanted to include libraries not in /usr/lib I'd specify
14 #   their path using -Lpath, something like:
15 LFLAGS = -lncurses
16
17 # define output directory
18 OUTPUT := output
19
```

Funciones Principales de Inicialización

- `initscr()`: Inicializa el terminal en modo de curses y prepara la ventana actual (llamada `stdscr`). Debe utilizarse antes que cualquier otra función de la librería.
- `refresh()`: La función `refresh()` comprueba la pantalla y actualiza sólo la parte que ha cambiado, mejorando el rendimiento.
- `endwin()`: Para liberar la memoria ocupada y terminar el uso de `ncurses` y dejar la terminal en el estado inicial.

Funciones Principales de Salida

Con Ncurses no debemos usar ni `cout` ni `cin` para la escritura y lectura de caracteres, sino que tendremos que usar las funciones de la librería.

- `addch()`: Para imprimir un caracter individual con atributos. Si un caracter está asociado a un atributo (negrita, subrayado, etc.), cuando curses imprime el carácter, este se imprime con ese atributo. Ej.: `addch(ch | A_BOLD | A_UNDERLINE);`
- `printw()`: Para imprimir salida con formato similar a `printf()`.
- `addstr()`: Para imprimir cadenas de caracteres con atributos.
- `move(fila, columna)`: Mueve el cursor a la enesima fila y enesima columna (primero es `Y` y luego es `X`).
- `mvaddch(fila, columna, ch)`: Mueve el cursor a la enesima fila y enesima columna e imprime un caracter.
- `mvprintw(f, c, string)`: Mueve el cursor y ejecuta `printw()`.
- `mvaddstr(f, c, string)`: Mueve el cursor y ejecuta `addstr()`.

Nota: Usar estas funciones es como ejecutar `move()` y luego alguna función de impresión.

Ejemplos de **printf()** o **printw()** en ncurses

Se implementa la función **printf(print formatted)**, para mostrar una cadena con formato: **printf("cadena %tipo", variable...)**

Tipo	Salida
%d	entero con signo (int) printf ("El numero entero es: %d", -10); Imprime: "El numero entero es: -10"
%f	Coma flotante decimal de precisión simple (float) printf ("el numero flotante es: %f", 3.1416); Imprime: "El numero flotante es: 3.141600"
%8.2f	tamaño total de 8 dígitos, con dos decimales printf ("el numero flotante es: %8.2f", 3.1416); Imprime: " 3.14" (con 7 espacios antes del 3).
%c	carácter (char) printf ("Caracteres: %c %c \n", 'a', 65); Imprime: "Caracteres: a A"
%s	cadena de caracteres (string) printf ("%s \n", "Esto es un string..."); Imprime: "Esto es un string..." y hace un cambio de párrafo

Funciones Principales de Entrada

- `getch()`: Para leer un solo caracter del teclado. Si utilizamos la función `cbreak()` podremos ingresar un caracter sin presionar *enter*. Y con la función `noecho()` el caracter no se mostrará en pantalla.
- `getstr()`: Para leer una cadena de caracteres.
- `scanw()`: Para obtener entrada con formato, similar a `scanf()`.
- `keypad(stdscr, valor)`: Activa o desactiva el uso de teclas especiales, F1, F2, ESC, flechas de dirección, etc. `valor = true`.
- `nodelay(stdscr, valor)`: Activa o desactiva (`true` o `false`) la espera para la lectura del teclado cuando se use `getch()`.
- `echo()/noecho()`: Con el `echo` activo lo que se escriba en el teclado se visualizará en la pantalla, con `noecho` esto no ocurre.
- `cbreak()/nocbreak()`: Activa o desactiva el envío inmediato de teclas sin la necesidad de presionar la tecla *enter*.
- `raw()/noraw()`: Es igual al anterior pero varía el tratamiento que hace de `ctrl+z` y `ctrl+c`.

Ejemplos de `scanf()` o `scanw()` en ncurses

Se implementa la función `scanf(scan-format, analizar con formato)`, para capturar una cadena con formato: `scanf("%tipo", &variable)`

Tipo	Ejemplo
%d	<i>// Este ejemplo guarda un número en n.</i> <pre>int n; printf("Introduce un numero: "); scanf("%d",&n);</pre>
%f	<i>// Este ejemplo guarda dos valores decimales.</i> <pre>float decimal1, decimal2; printf("Introduce dos números decimales separados con un espacio:\n"); scanf("%f %f", &decimal1, &decimal2);</pre>
%c	<i>// Este ejemplo guarda un caracter en m.</i> <pre>char m; printf("Introduce un caracter: "); scanf("%c",&m);</pre>
%s	<i>// Este ejemplo guarda una cadena de caracteres, // solamente una palabra, en cad. Nótese la ausencia de &amp.</i> <pre>char cad[20]; printf("Introduce una palabra: "); scanf("%s",cad); printf("Introduce una palabra: "); <i>// Lee máximo 10 caracteres y le concatena el caracter cero.</i> scanf("%10s",cad);</pre>

Atributos y Colores

Los atributos son formatos que pueden tener los caracteres. Ya vimos como podemos utilizar `addch()` para imprimir un caracter con atributos: `addch('A'|A_BOLD|A_UNDERLINE);`

Atributo	Descripción
A_NORMAL	Visualizacion normal (sin resaltar)
A_STANDOUT	Mejor modo del terminal para resaltado
A_UNDERLINE	Subrayado
A_REVERSE	Video inverso
A_BLINK	Intermitente
A_DIM	Mitad de brillo
A_BOLD	Brillo extra o negrita
A_PROTECT	Modo protegido, no puede ser sobrescrito
A_INVIS	En modo Invisible o en blanco

Constante	Valor
COLOR_BLACK	0
COLOR_RED	1
COLOR_GREEN	2
COLOR_YELLOW	3
COLOR_BLUE	4
COLOR_MAGENTA	5
COLOR_CYAN	6
COLOR_WHITE	7

Colores

Para poder utilizar los colores de la terminal debemos llamar a la función `start_color()`. Después de eso, podemos usar las capacidades de color de nuestra terminal utilizando distintas funciones. Para averiguar si una terminal tiene capacidades de color o no, podemos utilizar la función `has_colors()`, que devuelve `FALSE` si la terminal no soporta color.

Ncurses inicializa todos los colores admitidos por la terminal cuando llama a `start_color()`. Accedemos a ellos mediante constantes como `COLOR_BLACK`, etc. Para realizar un manejo efectivo de los colores, debemos definir pares. Los colores siempre se utilizan en pares. Esto significa hacer uso de la función `init_pair()` para definir el color de el frente y el fondo para un numero de pares deseado. Después los pares pueden ser utilizados como un atributo normal con la función `COLOR_PAIR()`.

Funciones Principales para los Colores

- `has_colors()`: devuelve TRUE o FALSE si la terminal tiene soporte para colores o no.
- `start_color()`: Inicia el modo para poder usar los colores.
- `init_pair(n, color del texto, color de fondo)`: Establece un par de colores, por ejemplo `init_pair(1, COLOR_RED, COLOR_YELLOW)` crea un par 1 con texto rojo y fondo amarillo.
- `attron()`: Activa el par de color, ej. `attron(COLOR_PAIR(1))`. También se puede combinar con algún atributo `attron(COLOR_PAIR(1) | A_UNDERLINE)`.
- `attroff()`: Desactiva el par de color, ej. `attroff(COLOR_PAIR(1))`.
- `inch()`: Devuelve el caracter y los atributos del mismo, a partir de la ubicación actual del cursor.
- `mvinch()`: Igual que el anterior, pero establecemos la posición del cursor a partir de donde se harán los cambios.

Cambio de Definiciones de Color

La función `init_color()` sirve para cambiar los valores RGB definidos por Ncurses inicialmente. Por ejemplo, si queremos aligerar la intensidad del color rojo para una minúscula, podemos utilizar esta función así:

```
init_color(COLOR_RED, 700, 0, 0);
```

```
/* param 1 : nombre de color
```

```
* param 2, 3, 4 : contenido RGB : min = 0, max = 1000 */
```

Nota: RGB es Red (rojo) Green (verde) Blue (azul)

Si la terminal no puede cambiar las definiciones de color, la función devuelve *error*. Podemos utilizar la función `can_change_color()` para saber previamente si la terminal tiene la capacidad de cambiar el contenido de color o no. El contenido rgb se escala de 0 a 1000. Inicialmente `COLOR_RED` es definido con un contenido de 1000(r), 0(g), 0(b). Podemos utilizar las funciones `color_content()` y `pair_content()` para conocer el contenido de color y la combinación de frente y fondo de un par.

Funciones Útiles

- `getmaxyx(stdscr, filas, columnas)`: Obtiene los límites de la terminal, devuelve los valores en las variables `filas` y `columnas`.
- `getyx(stdscr, y, x)`: Obtiene la posición actual del cursor, devuelve los valores en las variables `y` y `x`.
- `curs_set(valor)`: TRUE o FALSE para mostrar u ocultar el cursor.
- `box(stdscr, ch1, ch2)`: Crea un marco en la pantalla asignando el carácter `ch1` para las líneas verticales y `ch2` para las horizontales.
- `wborder(stdscr, ch1, ch2, ch3, ch4, ch5, ch6, ch7, ch8)`: Crea un marco en la pantalla asignando caracteres distintos para cada uno de los bordes y las esquinas.
- `mvhline(x, y, ch, c)`: Imprime una línea horizontal en la posición `x`, `y`, utilizando el carácter `ch` repitiéndolo `c` cantidad de veces.
- `mvvline(x, y, ch, c)`: Imprime una línea vertical en la posición `x`, `y`, utilizando el carácter `ch` repitiéndolo `c` cantidad de veces.