

# PRACTICA 2 – NCURSES

Alberto Socas Mendoza – PDIH

## Cuestiones para resolver

1. Instalar la librería ncurses, crear los programas de ejemplo ofrecidos más arriba, y comprobar su funcionamiento.
2. Crear un juego sencillo tipo “pong” partiendo del ejemplo del movimiento de la pelotita. Al iniciar el juego debe mostrarse un recuadro en una pantalla de bienvenida explicando los controles del juego. Tras una pausa se inicia el juego en sí mismo.

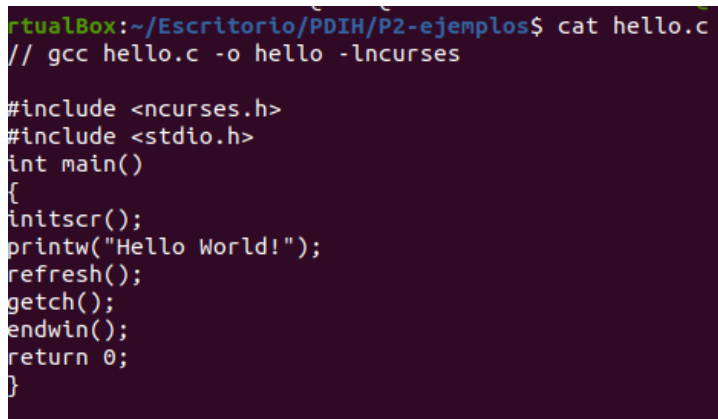
## Resueltos

Primero debemos instalar la librería ncurses en Linux, por lo que ejecutaremos el comando

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

tras ejecutarlo, ya deberíamos tenerlo instalado. Para comprobar dicha instalación vamos a ejecutar algunos programas.

Vamos a probar nuestro primer programa *hello.c*, que mostramos a continuación



```
rtualBox:~/Escritorio/PDIH/P2-ejemplos$ cat hello.c
// gcc hello.c -o hello -lncurses

#include <ncurses.h>
#include <stdio.h>
int main()
{
    initscr();
    printw("Hello World!");
    refresh();
    getch();
    endwin();
    return 0;
}
```

Ilustración 1

Para compilar este programa, ejecutamos la siguiente línea de código

***gcc hello.c -o hello -lncurses***

Tras ejecutarla, en la misma carpeta se nos crea un archivo ejecutable. Para comprobar su funcionamiento ejecutamos

***./hello***

Y el resultado del programa nos saldrá en consola. En este caso vemos que la consola ha impreso “Hello World!”

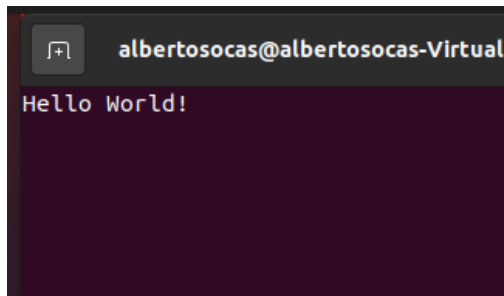


Ilustración 2

Vamos a seguir probando programas, el siguiente es *ventana.c*, aquí mostramos su contenido

```
// gcc ventana.c -o ventana -lncurses
//
#include <stdlib.h>
#include <ncurses.h>

int main(void) {
    int rows, cols;

    initscr();

    if (has_colors() == FALSE) {
        endwin();
        printf("Your terminal does not support color\n");
        exit(1);
    }

    start_color();
    init_pair(1, COLOR_YELLOW, COLOR_GREEN);
    init_pair(2, COLOR_BLACK, COLOR_WHITE);
    init_pair(3, COLOR_WHITE, COLOR_BLUE);
    clear();

    refresh();
    getnxyx(stdscr, rows, cols);

    WINDOW *window = newwin(rows,cols,0,0);
    wbkgd(window, COLOR_PAIR(3));
    box(window, '|', '-');

    mvwprintw(window, 10, 10, "una cadena");
    wrefresh(window);

    getch();
    endwin();
    return 0;
}
```

Ilustración 3

De nuevo, vamos a compilar el programa con

***gcc ventana.c -o ventana -lncurses***

Y seguidamente ejecutamos con

***./ventana***

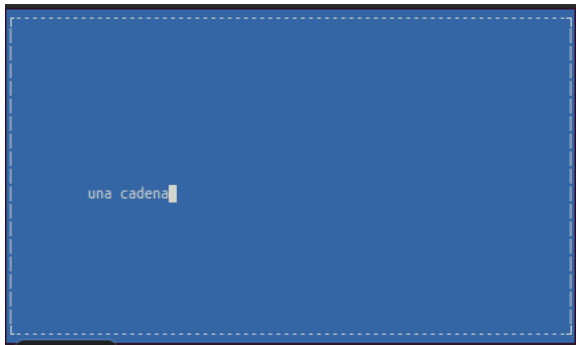


Ilustración 4

Por último, vamos a ejecutar el programa *pelotita.c*, cuyo código es

```
#include <ncurses.h>
#include <unistd.h>

#define DELAY 30000

int main(int argc, char *argv[]) {
    int x = 0, y = 0;
    int max_y = 20, max_x = 20;
    int next_x = 0;
    int direction = 1;

    initscr();
    noecho();
    curs_set(FALSE);

    while(1) {
        clear();
        mvprintw(y, x, "o");
        refresh();

        usleep(DELAY);

        next_x = x + direction;

        if (next_x >= max_x || next_x < 0) {
            direction *= -1;
        } else {
            x += direction;
        }
    }

    endwin();
}
```

Ilustración 5

Este programa “jugará” con una pelota denominada “o” en el eje x.

Lo compilamos con el siguiente código

***gcc pelletita.c -o pelletita -lncurses***

Y lo ejecutamos con el siguiente código

***./pelotita***

Nuestro resultado será la pelotita recorriendo el eje x de un lado a otro.

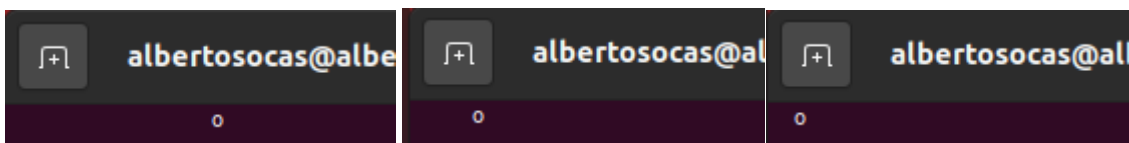


Ilustración 6

## PARTE 2. JUEGO PONG

Aquí añadiré unas capturas de pantalla del juego Pong creado en C. Además, añadiré el código en la entrega.

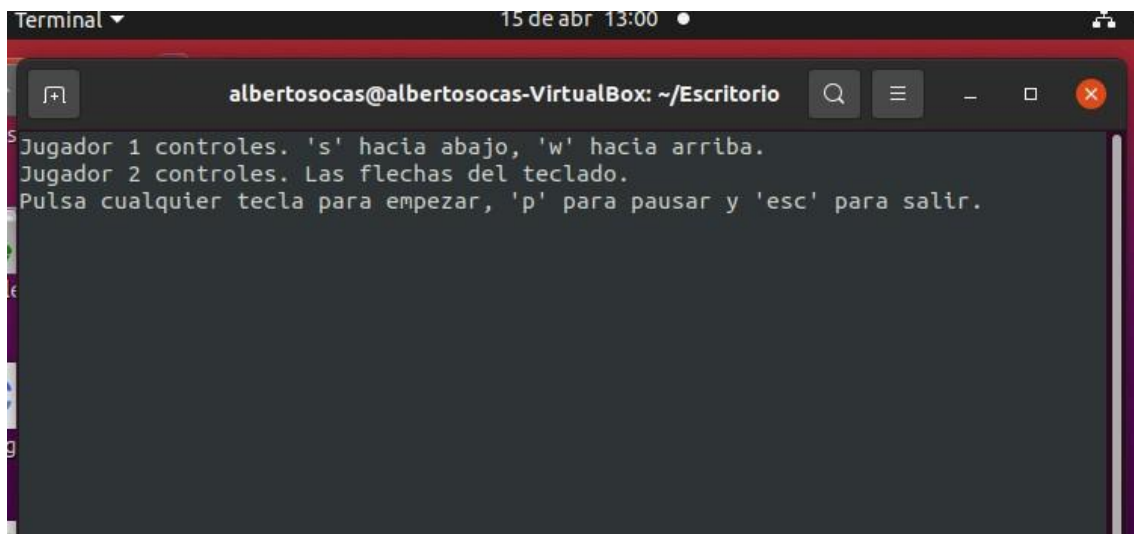


Ilustración 7

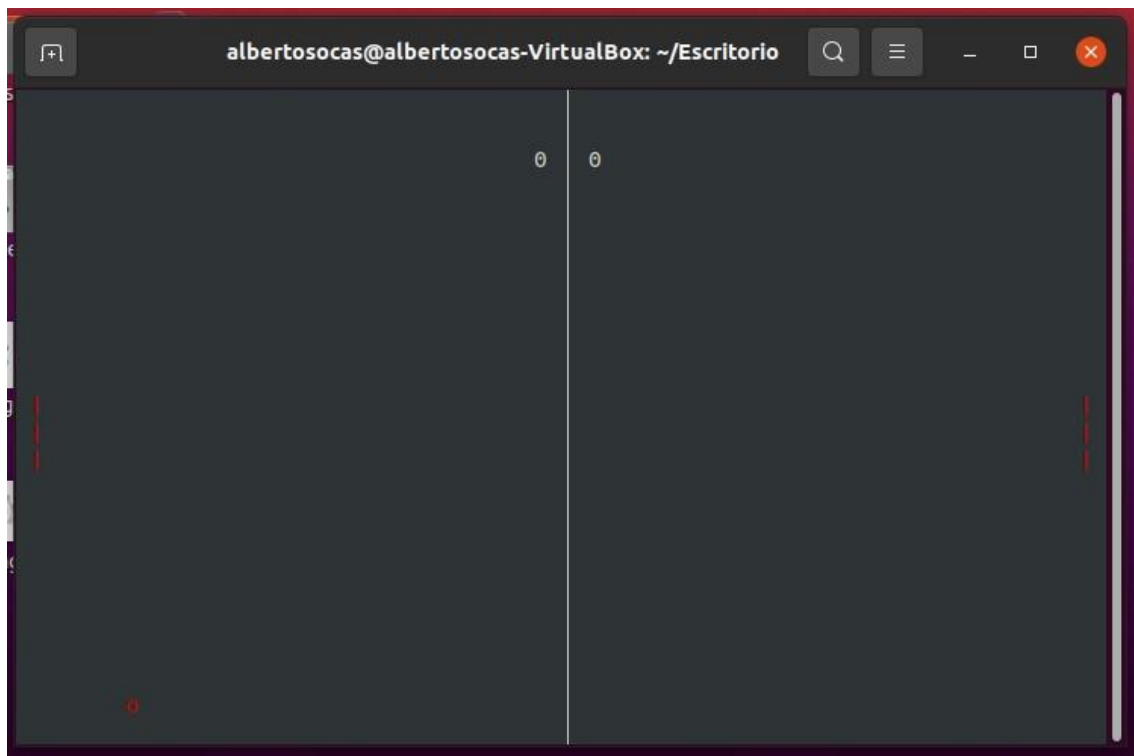


Ilustración 8