**Programación de puerto serie en Linux Ubuntu**

Creación de dos puertos series virtuales e interconectados para probar los ejercicios mediante el comando socat:

Instalación: sudo apt-get install socat

Creación de puertos:

Con el siguiente comando se van a crear dos puertos serie virtuales conectados el uno al otro, por lo tanto cuando se envíen datos por uno se van a poder leer en el otro y viceversa.

sudo socat -d -d -d -v -x PTY,link=/dev/ttyUSB0,mode=777,unlink-close,raw,echo=0 PTY,link=/dev/ttyS1,mode=777,unlink-close,raw,echo=0

screen: programa para monitorear y enviar datos por Puerto serie:

Instalación: sudo apt install screen

Abrir otras 2 terminales (diferentes a la utilizada para ejecutar el comando de socat) ejecutar:

Para monitorear ttyUSB0:

sudo screen /dev/ttyUSB0

Para monitorear ttyS1:

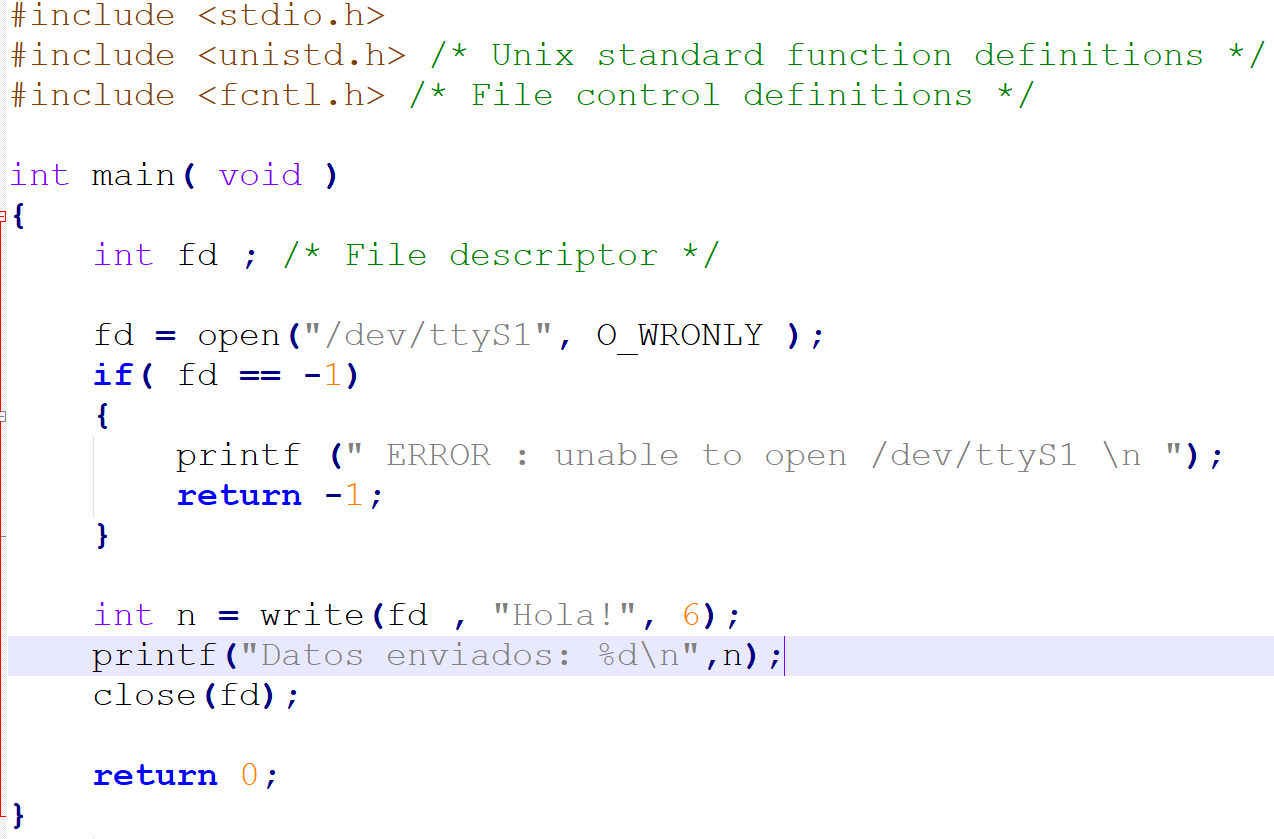
sudo screen /dev/ttyS1

Al ejecutar el comando screen la terminal se pondrá en blanco y va a mostrar los caracteres recibidor por el puerto serie correspondiente. Además, si se escriben caracteres mediante el teclado en esa terminal, no se mostrarán en pantalla, sino que se enviarán por el puerto serie. Cuando screen esta conectado a un puerto, este puerto no podrá ser escrito por otra aplicación, solo leído. Para cerrar la sesión de screen usar Ctrl +a y d o Ctrl + a luego Ctrl + d.

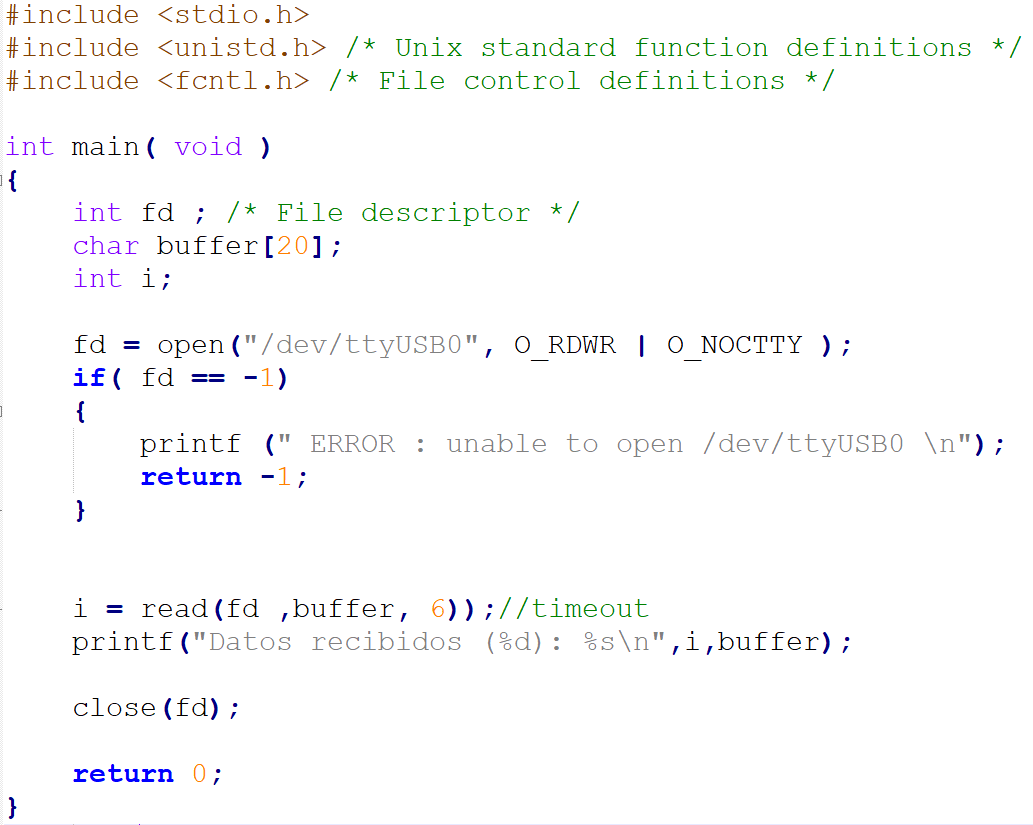
Ejercicios de programación en C:

Compilar y ejecutar los siguientes ejercicios en otra terminal, recordar que si screen esta conectado al puerto a utilizar en el programa este podrá fallar al no tener permiso de lectura o escritura.

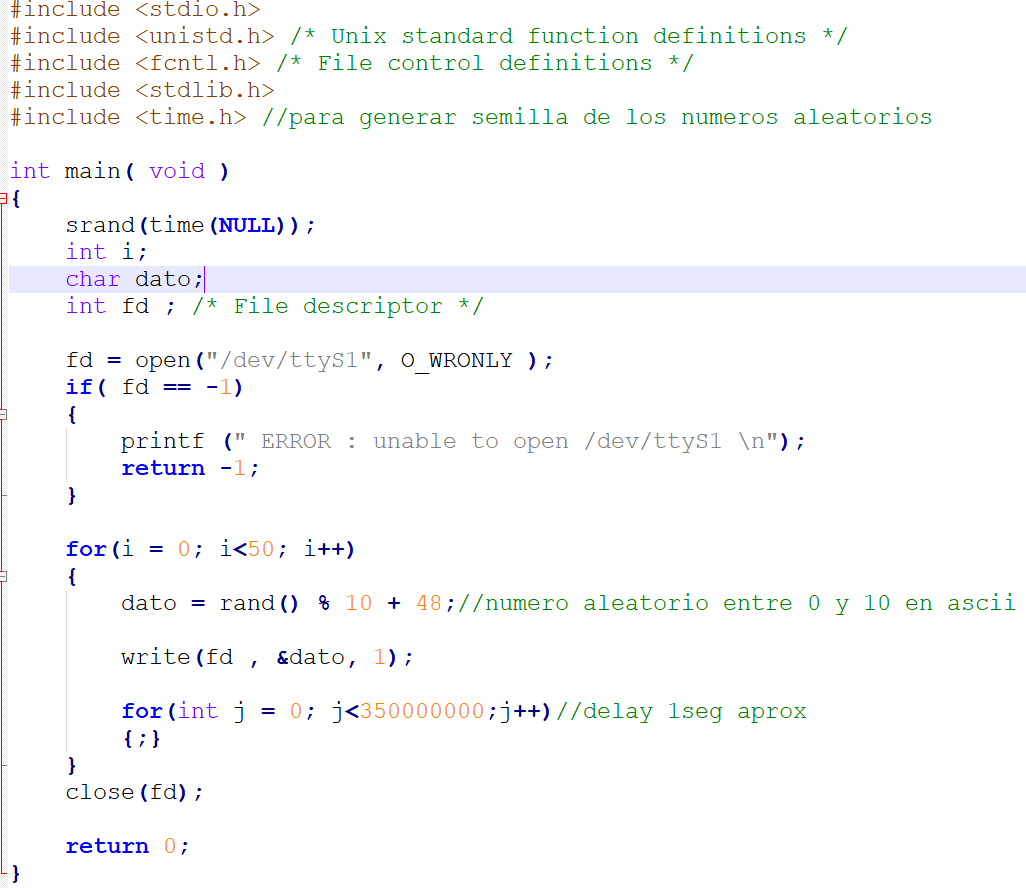
1. Escriba un programa en C que abra un puerto serie en modo escritura y envíe un mensaje de texto como “¡Hola Mundo!”.



1. Escriba un programa en C que abra un puerto serie en modo lectura y muestre en pantalla una cadena de caracteres recibida desde el puerto.



1. Escriba un programa en C que envíe por puerto serie un numero entero aleatorio cada 1 segundo.



1. Escriba un programa en C reciba números enteros desde el puerto serie, los guarde en un archivo de texto (uno por renglón) y que cada 10 números recibidos muestre en pantalla su promedio.
2. Modifique el programa 3 para que envíe en lugar del número aleatorio, la lectura de una tensión de 0 a 5v obtenida mediante un conversor analógico digital (AD) de 10bits de las siguientes maneras (pueden ser programas diferentes o uno mismo que solicite al usuario de qué manera enviar):
   1. En formato de texto: convertir el valor numérico con 2 posiciones decimales a cadena de caracteres con el formato “x,xx;”
   2. En formato binario de a un byte, utilizando la siguiente unión:

union dato {  
   byte b[4];  
   float fval;  
};

* 1. En formato binario directamente como float

Tener en cuenta que en este caso se considera que la lectura de la tensión por un conversor AD devuelve un valor entero sin signo de 10bits (los bits superiores no contienen datos validos) equivalente al valor analógico de la tensión medida. Es decir un 0 corresponde a 0 V y 210 a 5V.

1. Modifique el programa 4 para recibir los valores enviados por el programa 5.

A continuación un ejemplo de código para convertir un valor entero de 10bits a ASCII:

uint16\_t temp = 0; //variable temporal de 16bits

uint8\_t **Unidades** = 0; //variables de 8 bit para c/cifra

uint8\_t Decenas= 0;

uint8\_t Centenas=0;

uint8\_t Miles=0;

temp = ADC\_10bit\_Result;//lectura del conversor AD

Unidades = temp % 10; // Ej 1023 % 10 -> Resto -> 3

temp = temp/10; // Ej temp = 1023/10 ->102

Decenas = temp % 10; // Ej 102 % 10 -> Resto -> 2

temp = temp/10; // Ej temp = 102/10 ->10

Centenas= temp % 10; // Ej 10 % 10 -> Resto -> 0

Miles= temp / 10; // Ej 10 / 10 -> Cociente -> 1

//ASCII Conversion

Unidades += 0x30;

Decenas+= 0x30;

Centenas+= 0x30;

Miles+= 0x30;

En las últimas 4 variables se obtendrán las cifras con el correspondiente valor ASCII del número original

Cambiar la configuración del puerto serie

En los ejemplos anteriores solo se utilizó el puerto abriéndolo en modo lectora o escritura. En estos casos el puerto mantiene la última configuración definida por el sistema operativo o la última aplicación que lo haya utilizado (como por ejemplo en el comando socat mostrado al inicio). Si se desea cambiar la configuración se puede utilizar la estructura termios como se muestra en el ejemplo de código adjunto.