

## Laboratorio 4

## Primera Parte

**Inciso 9:** En este inciso se pedía que en la terminal de la Raspberry Pi se utilizara el comando de *raspi-config* para ver su funcionamiento. Al ingresar el comando como tal en la terminal salta un error y una sugerencia. Esto se da ya que normalmente cuando se abre la terminal se abre como un usuario, el usuario que normalmente usamos no tiene privilegios para cambiar cosas del sistema, y como el nombre del comando lo indica, vamos a configurar algunas cosas. Una analogía puede ser cuando se abre la terminal de Windows como administrador o como un usuario. Entonces, la sugerencia que nos da la terminal es escribir *sudo raspi-config* para poder acceder a las configuraciones. El comando sudo (superuser do) permite a un usuario ejecutar un comando ya sea como otro usuario o como el superuser. El superuser como vimos anteriormente se le conoce como *root* y tiene acceso al todo el sistema y puede realizar cualquier acción, hasta algunas que puedan afectar al sistema.

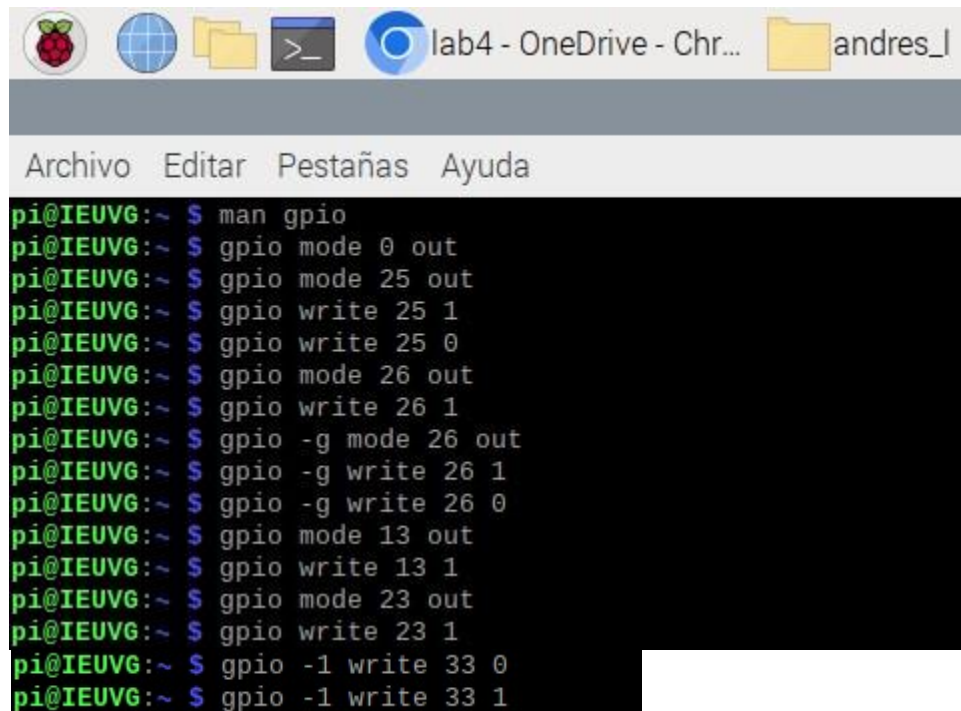
**Inciso 12:** Al compilar ambos programas todo funcionaba igual que en CentOS ya que el programa de ejemplo de threads, no reconocía la función para crear threads, entonces al compilar se tenía que incluir la librería como `-lpthreads`; esto sucedía en CentOS mas no en Cygwin donde no se tiene que especificar la librería. En cuanto a la ejecución de ambos programas, todos funcionaron de igual forma que en Cygwin y CentOS.

[illegible]

Figura 1. Ejemplos de fork y pthreads, y procesos.

## Segunda Parte

### Inciso 1:



The screenshot shows a terminal window on a Raspberry Pi. The window has a title bar with icons for the Raspberry Pi logo, a globe, a folder, a terminal icon, and a OneDrive icon labeled 'lab4 - OneDrive - Chr...'. Below the title bar is a menu bar with 'Archivo', 'Editar', 'Pestañas', and 'Ayuda'. The terminal content shows a series of commands being executed at the 'pi@IEUVG:~' prompt:

```
pi@IEUVG:~$ man gpio
pi@IEUVG:~$ gpio mode 0 out
pi@IEUVG:~$ gpio mode 25 out
pi@IEUVG:~$ gpio write 25 1
pi@IEUVG:~$ gpio write 25 0
pi@IEUVG:~$ gpio mode 26 out
pi@IEUVG:~$ gpio write 26 1
pi@IEUVG:~$ gpio -g mode 26 out
pi@IEUVG:~$ gpio -g write 26 1
pi@IEUVG:~$ gpio -g write 26 0
pi@IEUVG:~$ gpio mode 13 out
pi@IEUVG:~$ gpio write 13 1
pi@IEUVG:~$ gpio mode 23 out
pi@IEUVG:~$ gpio write 23 1
pi@IEUVG:~$ gpio -1 write 33 0
pi@IEUVG:~$ gpio -1 write 33 1
```

Figura 2. Comandos para escribir a un puerto digital de la raspberry pi.

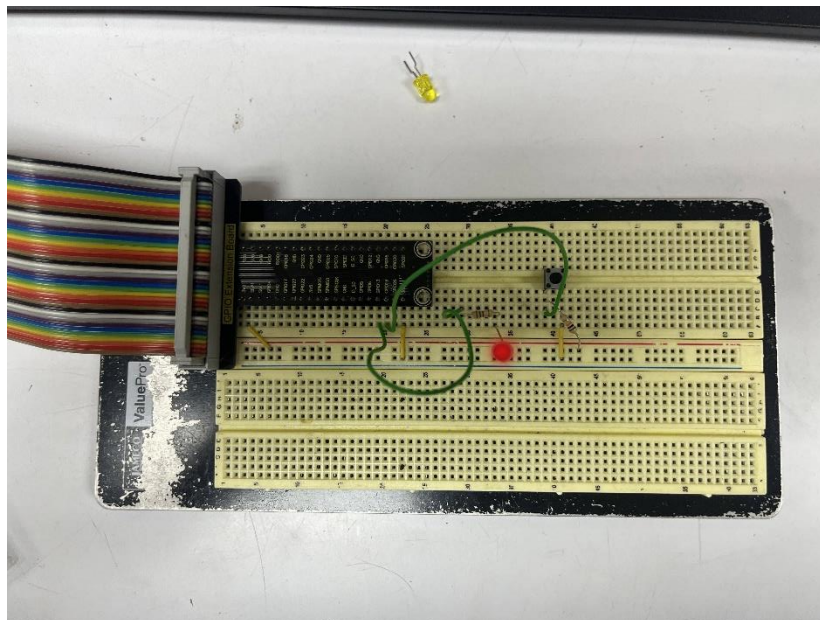


Figura 3. Protoboard con led encendido

Al observar la Figura 2 podemos ver las líneas de comando utilizadas, con el comando `gpio mode out` se puede definir como salida el puerto que utilizaremos como el led. También se puede ver que se puede agregar la opción de `-g`, `-1` o sin ninguna opción directamente. Esto cambia la numeración, si se utiliza sin opción es que se usará la numeración de wiringPi, si se utiliza `-g` se usará la numeración de BCM; y por último si se utiliza `-1` se utiliza la numeración física de los pines. Al usar el comando `write` y especificar el pin podemos mandarle un 0 o 1 lógico.

## Inciso 2:

```
pi@IEUVG:~ $ gpio mode 24 input
pi@IEUVG:~ $ gpio mode 24 in
pi@IEUVG:~ $ gpio read 24
1
pi@IEUVG:~ $ gpio read 24
0
pi@IEUVG:~ $ gpio -g read 19
1
pi@IEUVG:~ $ gpio -g read 19
0
pi@IEUVG:~ $ gpio mode 24 up
pi@IEUVG:~ $ gpio read 24
1
pi@IEUVG:~ $ gpio read 24
0
pi@IEUVG:~ $ gpio -g read 19
1
pi@IEUVG:~ $ gpio -g read 19
0
pi@IEUVG:~ $ gpio -1 read 35
1
pi@IEUVG:~ $ gpio -1 read
Usage: gpio read pin
pi@IEUVG:~ $ gpio -1 read 35
0
pi@IEUVG:~ $
```

Figura 4. Comandos para leer de un puerto digital de la raspberry pi.

En la figura 4 se puede observar algunos comandos para leer un puerto de la raspberry pi. Para elegir un pin como entrada, se puede realizar como `gpio mode [pin] in`; aunque también se puede utilizar `up` y `down` para especificar si se quiere una resistencia pull-up o pull-down. De la misma forma que para escribir, se puede elegir la numeración que se utiliza con las opciones `-g`, `-1` o sin ninguna opción. Al analizar lo que devuelve cuando se realiza una lectura vemos que cuando el botón está sin presionar, la lectura devuelve un 1, y cuando se lee mientras está presionado devuelve un 0. Esto es porque se conectó el botón utilizando la configuración de pull-up, que devuelve 1 lógico mientras no se presiona, y 0 lógico mientras está presionado.

## **Referencias**

Team GeekforGeeks. (2023). sudo command in Linux with Examples.  
<https://www.geeksforgeeks.org/sudo-command-in-linux-with-examples/>