

Trabajo práctico 2

Sistemas Operativos de Propósito General. CESE

Objetivo:

Se deberá desarrollar un sistema de domótica para el manejo de 3 luces. El sistema permitirá controlar las luces (emuladas con leds) desde una pagina web y desde los pulsadores del controlador.

Partes del sistema:

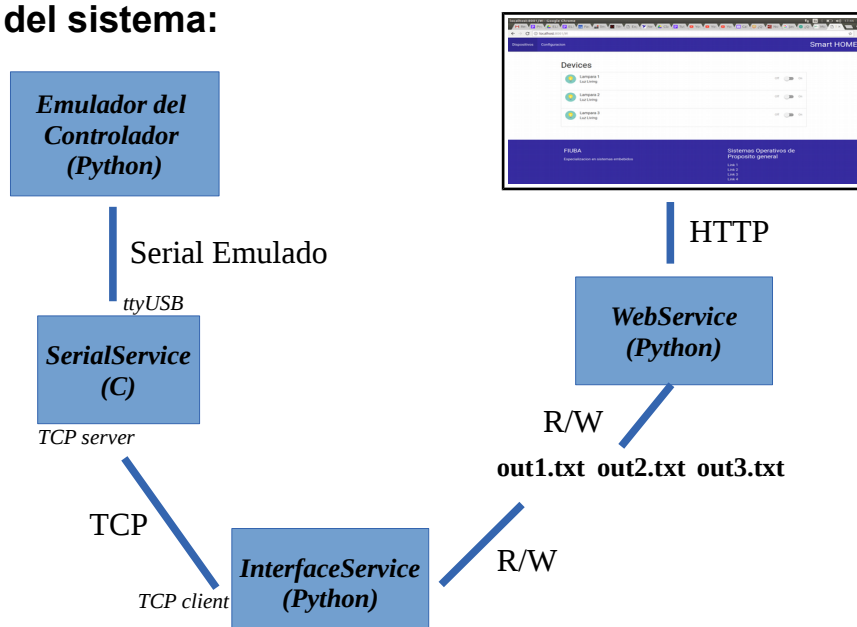
Emulador:

Mediante un script en python se simulará un dispositivo de hardware controlador de las 3 luces y las 3 teclas que encenderán o apagarán dichas luces. Este emulador permite simular el uso del puerto serie para enviar tramas para controlar el estado de las salidas y recibir tramas cuando se presionen los pulsadores. En este caso, los pulsadores no existen sino que el mismo emulador permitirá ingresar eventos de pulsado desde la consola utilizando el teclado.

PC:

Mediante la pc se alojará un sitio web también provisto, el cual mostrará una lista de lamparas y un botón que permitirá seleccionar su estado (encendida o apagada).

Arquitectura del sistema:



Trabajo práctico número 2

El servidor web leerá y escribirá los archivos : **/tmp/out1.txt, /tmp/out2.txt y /tmp/out3.txt**

- El servidor web lee el contenido de los archivos para mostrar el estado de las lámparas en la página web.
- El servidor web escribe el contenido de los archivos al presionar el botón para cambiar la lámpara de estado.
- El servicio *InterfaceService* escribirá en los archivos (en la salida lámpara correspondiente) cuando reciba por socket una trama de que se presionó un pulsador.
- El servicio *InterfaceService* leerá los archivos y al detectar un cambio enviará por socket una trama indicando el nuevo valor de la lámpara.

Servicio “Serial Service”

El emulador se comunicará con el servicio *SerialService* mediante el puerto serie, que deberá recibir las tramas indicando que se presionó un pulsador. Cuando esto ocurra, deberá informarlo al servicio *InterfaceService* mediante socket TCP.

También deberá enviar por el puerto serie la trama que le indica al emulador si debe encender la salida 1, salida 2 o salida 3, información que le provee el servicio *InterfaceService*. (Cuando se modifican los archivos desde el sitio web)

Por último este servicio iniciará un servidor TCP para que el servicio *InterfaceService* se pueda conectar y comunicar.

Servicio “Interface Service”

Este servicio creará un cliente TCP que se conectará al servicio *SerialService*. Se encargará de leer y escribir en los archivos mencionados previamente para que la información se vea reflejada en la página web.

Funcionamiento

Cuando se presione un pulsador desde el emulador (utilizando la consola y el teclado), ésta lo informará por puerto serie al *SerialService*, el cual deberá enviarlo mediante un paquete TCP al *InterfaceService* para que éste escriba el nuevo valor de la lámpara en el archivo correspondiente.

Cuando se cambie desde la página web el estado de una lámpara, el servidor web escribirá en el archivo correspondiente el nuevo valor, el cual lo detectará el *InterfaceService* y lo informará mediante un paquete TCP al *SerialService*, el cual enviará la trama por el puerto serie al emulador, para que ésta modifique el estado de la salida correspondiente.

Lo que se provee

- 1) Emulador del controlador
- 2) Página web
- 3) Servidor web
- 4) Servicio “InterfaceService”
- 5) Biblioteca para el puerto serial simulado

Lo que se debe desarrollar

- 1) Servicio “SerialService”

Protocolo serie entre el Emulador y SerialService y protocolo TCP entre SerialService e InterfaceService

Seteo encendido de salida (hacia el emulador)

“>OUT:X,Y\r\n”

Siendo X el número de salida (0,1, o 2) e Y el estado (0 o 1)

Esta trama es la misma que enviará InterfaceService a SerialService.

Evento pulsador (desde el emulador)

“>SW:X,Y\r\n”

Siendo X el número de salida (0,1, o 2) e Y el nuevo estado a setear (0 o 1)

Esta trama es la misma que enviará SerialService a InterfaceService.

Puesta en marcha del sistema

- 1) Inicializamos el programa *InterfaceService*. Abrir una terminal.

```
cd TP2
cd InterfaceService
python Main_python3.py
```

- 2) Inicializamos el servidor web. Abrir una terminal.

```
cd TP2
cd web
./runServer_python3.sh
```

- 3) Abrimos un navegador y en la url ponemos **http://localhost:8001** . Deberá aparecer la página web.

- 4) Inicializamos el emulador:

```
cd TP2
cd EmuladorHardware
python Emulador.py
```

- 5) Abrir el programa “SerialService” desarrollado asignándole la ttyUSB de la placa.

Consideraciones a tener en cuenta en el programa a desarrollar

- 1) La función que lee datos del puerto serie no es bloqueante.
- 2) La función que lee datos del cliente tcp es bloqueante. No cambiar este comportamiento.
- 3) Dadas las condiciones de los puntos 1 y 2, se recomienda lanzar un thread para manejar la comunicación con el cliente TCP y otro (opcional, no necesario) para la comunicación con el puerto

Trabajo práctico número 2

serie.

- 4) No generar condiciones donde el uso del CPU llegue al 100% (bucles sin ejecución bloqueante).
- 5) El programa debe soportar que el cliente se desconecte, se vuelva a conectar y siga funcionando el sistema.
- 6) El programa debe poder terminar correctamente si se le envía la signal SIGINT o SIGTERM.
- 7) Se debe considerar sobre qué hilo de ejecución se ejecutarán los handlers de las signals.
- 8) Los temas que deberán implementarse en el desarrollo son los siguientes:

- Sockets
- Threads
- Signals
- Mutexes (De ser necesario)

Datos adicionales

Versión de python: 3+

Puerto TCP : 10000

IP local : 127.0.0.1

Puerto y baudrate para ttyUSB para el emulador : 1 y 115200 . Para abrir el puerto serie simulado desde SerialService se deberá realizar la invocación de *serial_open* como sigue:

```
if(serial_open(1,115200)!=0)
{
    printf("Error abriendo puerto serie\r\n");
    ...
}
```

Condiciones de aprobación

Para considerar el trabajo práctico aprobado, el mismo deberá encontrarse funcionando con todos los controles de error implementados, una mínima documentación con comentarios y el código deberá encontrarse bien indentado y legible, además de cumplir con las consideraciones de desarrollo detalladas previamente.

El trabajo deberá ser terminado y entregado al finalizar la clase 8 y se considerará instancia de recuperatorio si se entrega hasta 3 días después. Luego de esa fecha se considerará al alumno desaprobado en la instancia de recuperación y deberá recursar la materia.