

Trabajo práctico 2

Sistemas Operativos de Propósito General. CESE. 2018

Objetivo:

Se deberá desarrollar un control de acceso por código numérico con la EDU-CIAA. El usuario deberá ingresar mediante las 4 teclas de la placa, un código de 4 dígitos (las teclas corresponden a los números del 1 al 4) y si el usuario tiene permisos, se le dará acceso (se simula encendiendo el led1 durante 5 segundos). El sistema tendrá una página web en donde podrá controlar en qué fecha y hora se realizó cada acceso (indicando ID ingresado, fecha y hora en una tabla). Para ello el sistema deberá escribir en un archivo de log todos los accesos.

Partes del sistema:

EDU-CIAA:

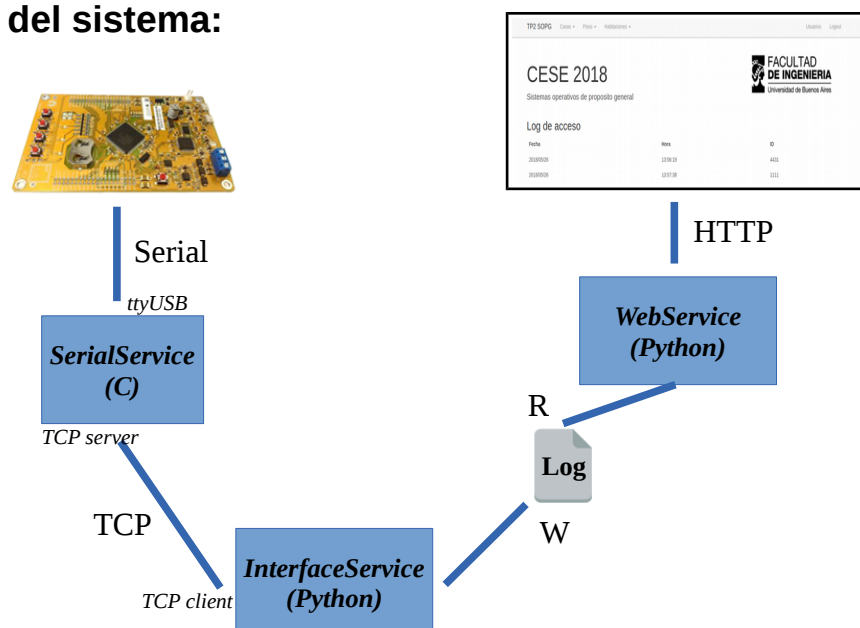
Mediante una placa EDU-CIAA se simulará un teclado con solo 4 teclas (números del 1 al 4) y una salida a relé con el led1. La placa contendrá un firmware provisto, que permite almacenar el número que ingresó el usuario y enviarlo por el puerto serie del conector usb de debug. Por el mismo puerto serie la placa recibirá un comando que activará el led1 durante cierto tiempo.

El firmware provisto está escrito utilizando la biblioteca firmwareV2 y se provee funcionando.

PC:

Mediante la pc se alojará un sitio web también provisto, el cual mostrará el log de los accesos que se realizaron, indicando fecha, hora e ID ingresado. La PC estará conectada a la placa mediante el puerto serie-usb de debug de la EDU-CIAA.

Arquitectura del sistema:



Trabajo práctico número 2

El servidor web leerá de un archivo: **/tmp/log.txt**

- El servidor web lee el contenido de este archivo para mostrar el log en la página web.
- El servicio *InterfaceService* escribirá en dicho archivo datos cuando sea notificado de un nuevo acceso.

Servicio “Serial Service”

La EDU-CIAA se comunicará con el servicio *SerialService* mediante el puerto serie, que deberá recibir las tramas de la placa indicando que se ingresó un código. Cuando esto ocurra, deberá informarlo al servicio *InterfaceService* mediante socket TCP.

También deberá enviar por el puerto serie la trama que le indica a la placa si debe encender el led1, información que le provee el servicio *InterfaceService*. (Cuando el código ingresado por el usuario es válido)

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 escribir el archivo de log mencionado previamente para que la información se vea reflejada en la página web y será quien decida si se le da o no acceso al ID ingresado.

Funcionamiento

Cuando se ingrese un código en la placa, ésta lo informará por puerto serie al *SerialService*, el cual deberá enviarlo mediante un paquete TCP al *InterfaceService* para que éste lo escriba en el archivo de log, y el nuevo ingreso se vea reflejado en la web cuando el servidor web sea consultado nuevamente por la página web. El *InterfaceService* provisto da acceso a cualquier ID ingresado.

Lo que se provee

- 1) Firmware de la EDU-CIAA
- 2) Página web
- 3) Servidor web
- 4) Servicio “InterfaceService”
- 5) Biblioteca para el puerto serial

Lo que se debe desarrollar

- 1) Servicio “SerialService”

Protocolo serie entre EDU-CIAA y SerialService y protocolo TCP entre SerialService e InterfaceService

Seteo encendido de led (hacia la edu-ciaa)

“>OUT:1\r\n”

La placa contesta “>OK\r\n”

Evento ingreso de código de 4 dígitos (desde la edu-ciaa)

“>ID:XXXX\r\n”

Puesta en marcha del sistema

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

```
cd TP2
cd InterfaceService
python Main.py
```

2) Inicializamos el servidor web. Abrir una terminal.

```
cd TP2
cd web
./runServer.sh
```

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

4) Conectamos la EDU-CIAA a la PC. Obtener la ttyUSB que se generó ejecutando:

```
ls /dev/ttyUSB*
```

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.

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 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

Puerto TCP : 10000

IP local : 127.0.0.1

Velocidad ttyUSB de la placa : 115200

Para grabar el firmware de la EDUCIAA:

Bajar el firmware “firmware_v2_TP2.tar.gz” y descomprimirlo. Luego:

```
cd firmware_v2
make clean
make
make download
```

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.