

11. ANEXO II. Configuración del sistema.

Este capítulo contiene un texto escrito durante el desarrollo del módulo de configuración a modo de HOWTO (cómo) para configurar la aplicación OperAIT. Para un usuario que sólo desee aprender a configurar el mismo, este texto resultará más sencillo y directo que leer el capítulo 6, aunque se recomienda consultar ambos, así como las excepciones comentadas en la página 64.

11.1. Introducción.

La configuración del sistema se realiza mediante un documento XML²³, cuyo nombre de archivo es `operait.conf`. En cada configuración se buscará un archivo con este nombre primero en el directorio desde el que se arrancó el programa y, si no lo encuentra, en el directorio `/etc`.

La configuración del sistema se carga en memoria al iniciar el programa, y en cualquier momento puede ser alterada durante la ejecución del mismo sin necesidad de reiniciarlo. Sencillamente hay que enviar la señal `HUP` al `PID` correspondiente. Tanto si la reconfiguración se realizó de forma correcta como si no se escribirá una línea en el fichero de traza indicado en la configuración antigua. Si hubo algún error en la configuración al arrancar el programa, se indicará el mismo por la salida de errores.

11.2. Estructura del archivo.

Todos los datos de configuración están contenidos en la marca XML `configdoc`, con la que debe comenzar (y terminar con `</configdoc>`) el archivo de configuración.

Los datos de configuración están divididos en tres secciones, de las cuales dos de ellas son **obligatorias** (si no se incluyen las secciones o campos obligatorios, el archivo se considera erróneo). El orden de aparición de dichas secciones no importa, pero cada una de ellas debe aparecer una vez como máximo. Las secciones son:

²³ En adelante, cuando se escriba "datos DDD contenidos en la marca XML X" corresponderá en el archivo XML con la entrada: `<X>DDD</X>`. Se permitirán espacios (que se obviarán) antes y después de DDD.

11.2.1. Opciones de archivo (fileoptions).

Es una sección **obligatoria** que está contenida en la marca XML `fileoptions`. Todos sus subcampos son a su vez obligatorios. Los nombres de las marcas correspondientes a estos campos son:

- `tracefile`: indica el nombre del archivo en el que se guarda toda la información de monitorización (puede incluirse una ruta directa o relativa a partir del directorio inicial desde donde se arrancó el programa).
- `wavdir`: nombre del directorio donde se almacenan los archivos de sonido. Al igual que antes, puede ser una ruta directa o relativa.
- `wavinvalidopt`: nombre del archivo de audio que escuchará el usuario cuando teclee una opción no válida.
- `wavpress`: nombre del archivo de audio para el mensaje “Pulse”.
- `wav0` hasta `wav9`: archivos de audio para los números de 0 a 9.
- `wava`: archivo de audio para *.
- `wavb`: archivo de audio para #.

Todos los archivos de audio deben ser archivos WAV en PCM de 16 bits a 8 KHz y monoaurales. Para grabar estos archivos, si se tiene una tarjeta de sonido instalada con un micrófono operativo, puede utilizarse el programa `rec` incluido con el paquete `gnu SoX` (Sound eXchange). La orden para ello es:

```
rec -r 8000 -s w -c 1 nombre_de_archivo.wav
```

11.2.2. Opciones de traza (traceoptions).

Esta es una sección **opcional**, cuyos subcampos son a su vez opcionales. El nombre de la etiqueta XML es `traceoptions`. Dentro de esta etiqueta, la aparición de un subcampo indica que se realiza la monitorización asociada a ese subcampo. Los subcampos, junto con la monitorización asociada a cada uno, son:

- `progstarttime`: fecha y hora de inicio del programa
- `connstarttime`: ídem para el inicio de cada conexión
- `connendtime`: ídem para el fin de cada conexión
- `keyboption`: opción de teclado introducida por el usuario, indicando fecha y hora de la misma y el número de conexión asociada de las dos posibles.
- `reconftime`: fecha y hora de cada intento de reconfiguración.

Estos campos deben estar vacíos, es decir, no debe haber ningún carácter entre el inicio y el fin de la etiqueta XML. Para ello puede usarse la sintaxis: `<keyboption/>`. Con una sola

marca XML indicamos el inicio y el fin de un campo vacío. En este caso se desea monitorizar las opciones de teclado introducidas por el usuario.

11.2.3. Menú de usuario (menutree).

Es un campo **obligatorio** con subcampos opcionales, pero que al menos debe contener algún subcampo dentro, es decir, no se permite un menú vacío. El nombre de la etiqueta XML es `menutree`. Los subcampos (**opcionales**) son:

- `snd`: debe contener un nombre de archivo WAV que se reproducirá por la red y que estará almacenado en el directorio indicado en la sección `fileoptions`, en el subcampo `wavdir`. Por ejemplo, para indicar que debe reproducirse el archivo de nombre “misonido.wav” esto sería: `<snd>misonido.wav</snd>`
- `option`: indica un submenú al que se accede mediante la pulsación de una combinación de teclas del teléfono. Puede contener a su vez todos los subcampos de `menutree`, creando por lo tanto en su interior un subárbol de menú al que se accede mediante dicha combinación de teclas. Dicho submenú no puede estar vacío, al igual que ocurre con el menú principal. Esta etiqueta XML tiene dos atributos:
 - `code`: indica cual es la combinación de teclas con la que se accede a dicho submenú. Es un atributo **obligatorio**.
 - `label`: indica un nombre de etiqueta que más tarde se podrá referenciar mediante el subcampo `goto`. Es **opcional**.

Si queremos crear un submenú al que se acceda mediante la pulsación de la tecla “#” y con una etiqueta asociada de nombre “misubmenu”, la sintaxis debe ser:

```
<option code="5" label="misubmenu">
  <!-- Aquí debe ir el cuerpo del submenú -->
</option>
```

- `redir`: debe contener un número de teléfono al que se transferirá la llamada que se atiende cuando el usuario llegue a este nodo. Por ejemplo, para desviar la llamada al número 23 se utilizaría: `<redir>23</redir>`.
- `goto`: sus datos son un nombre de etiqueta a partir del cual el programa debe seguir procesando el árbol de menú. Si, por ejemplo, deseamos que el programa siga procesando el árbol a partir del submenú con etiqueta “misubmenu”, debemos escribir: `<goto>misubmenu</goto>`.

11.2.4. Funcionamiento del programa.

El funcionamiento del programa de cara al usuario se basa en esta serie de menús anidados y se describe a continuación.

El programa comienza a procesar el árbol por la primera entrada que encuentra tras `menutree`. Cada entrada puede ser:

- `snd`: En tal caso, se comienza a volcar los datos desde el archivo indicado a la red, y se deja de procesar hasta que se terminan dichos datos o hasta que el usuario pulsa una tecla.
- `option`: primero se envían por la red los datos asociados al mensaje “pulse”, que deben estar almacenados en el fichero indicado en el subcampo `wavpress` de la sección `fileoptions`. Luego, y en función del atributo `code`, se reproducirán los datos asociados a los mensajes “uno”, “dos”, etc., que deben estar almacenados en los ficheros indicados en las marcas `wavX` (con X desde '0' a '9', 'a' o 'b') de la sección `fileoptions`. De esta forma, el usuario escuchará el mensaje “Pulse uno nueve” si introducimos la siguiente configuración:

```
<option code="19">
  <!-- Cualquier contenido -->
</option>
```

- `redir`: si el programa llega a procesar una entrada de este tipo, se realizará una transferencia de llamada de la conexión actual al número indicado.
- `goto`: si se encuentra una entrada `goto`, el programa seguirá procesando el árbol a partir del submenú que tenga asociada la etiqueta coincidente.

Cuando se llegue al final de un (sub)menú, se espera unos segundos y se vuelve a empezar por el principio del mismo. Esto se repite hasta tres veces y se corta la conexión si el usuario no cambia de submenú.

Si en algún momento el usuario pulsa alguna tecla, ésta se almacena en la memoria junto otras teclas pulsadas anteriormente y se comprueba si este conjunto de teclas coincide con el atributo `code` de algún campo `option` del (sub)menú actual. Si es así, se sigue procesando la primera entrada del submenú asociado a dicho campo `code`. Si no hay ninguna posibilidad de coincidencia en ningún campo `option` se comienza a reproducir el archivo indicado en el campo `wavinvalopt` de la sección `fileoptions`.

Esto quiere decir que si sólo hay un campo `option` con el valor “19” en el atributo `code` y el usuario sólo ha pulsado la tecla “1”, el programa no cambiará el submenú ni indicará error hasta que no se pulse otra tecla más, y seguirá procesando las entradas del (sub)

menú actual. El programa no comprueba que haya dos campos `option` con mismo atributo `code` dentro de un mismo (sub)menú, es el creador del archivo de configuración el que debe evitar esto, pues si ocurre será el primer caso que aparezca en el (sub)menú el que siempre se seleccione²⁴.

11.3. Ejemplo

El ejemplo original ha sido incluido en el apartado 6.3.1(“Caso práctico de configuración”), en la página 42.

²⁴ Consultar el apartado 7.3.3 para comprobar otra serie de excepciones para los valores del campo `code`.

