

Trabajo Práctico 3 - Sistemas Embebidos

Nuevamente la Provincia de Córdoba le solicita a la Escuela de Ingeniería en Computación de la FCEfYN-UNC, que realice el diseño, desarrollo y prueba del sistema de gestión y acceso a una base de datos hidro-meteorológica.

Se solicita que, a partir de lo realizado en el Trabajo Practico Nº1, utilizando el mismo set de datos, y la misma placa de desarrollo, se desarrolle lo siguiente:

- 1) Se realice un estudio de las distintas implementaciones web servers disponibles para sistemas embebidos, realice una comparación y justifique la selección de uno de ellos, que deberá instalar en el sistema operativo, y que deberá ejecutarse automáticamente cada vez que este se reinicia el sistema embebido.
- 2) Sobre el servidor web, debe desarrollarse una interfaz web simple, con múltiples páginas, donde cada una debe mostrar, utilizando HTML, CGI y Pearl o C, los siguientes requerimientos:
 - a) Pagina que reporte información sobre recursos varios del sistema embebido:
 - a. Procesador
 - b. Memoria
 - c. Uptime
 - d. Fecha y Hora actual
 - b) Pagina que permita ejecutar los comando desarrollados para el TP1, sin la utilización de sockets y con las siguientes modificaciones:
 - a. **descargar**: descarga un archivo con todos los datos de una nº_estación seleccionable.
 - b. **diario_precipitación**: permite seleccionar una nº_estación (de las disponibles), y muestra en pantalla el acumulado diario de la variable precipitación (*nº_día: acumulado mm*).
 - c. **mensual_precipitación**: permite seleccionar una nº_estación (de las disponibles) y muestra en pantalla el acumulado mensual de la variable precipitación (*nº_día: acumulado mm*).
 - d. **promedio**: permite seleccionar una variable (de las disponibles), y muestra el promedio de todas las muestras de la variable seleccionada de cada estación (*nº_estacion: promedio*).
 - e. NO implementa **connect** y **desconectar** (dado que no estamos trabajando con sockets).
 - c) Página que liste los módulos instalados en el sistema, y que posea un formulario que permita subir un archivo al servidor, controlar que este sea un archivo válido (del tipo módulo), e instalarlo en el kernel del sistema operativo. También debe poseer un botón para removerlo. Explicar todo el proceso.

- d) Desarrollar un módulo (driver) simple y vacío, que sólo imprima "Hello World" al instalarse y "Good ByeWorld" al ser removido del kernel. Este será el módulo que se debe instalar en el punto anterior.

Se debe entregar:

- a) Informe con el esquema dado en clase, que incluya una guía al estilo "how to" de cómo se realizó el trabajo (paso por paso).
- b) Todo código de fuente desarrollado y el binario del driver, con cualquier instructivo extra que crea necesario, Makefile, documentación, etc.
- c) Una imagen del sistema desarrollado.

Se recomienda el uso de Cppcheck y la compilación con el uso de las flags de warning -*Werror*, -*Wall* y -*pedantic*. Se pide utilizar el estilo de escritura de código de GNU [1] o el estilo de escritura del kernel de Linux [2].

Se debe asumir que las pruebas de compilación se realizarán en un equipo que cuenta con las herramientas típicas de consola para el desarrollo de programas (Ej: gcc, make), y **NO** se cuenta con herramientas "GUI" para la compilación de los mismos (Ej: eclipse).

FECHA DE ENTREGA: domingo, 28 de mayo de 2017, 23:55

Referencias y ayudas:

- a) <http://www.linuxdevcenter.com/pub/a/linux/2007/07/05/devhelloworld-a-simple-introduction-to-device-drivers-under-linux.html?page=1>
- b) <http://www.oreilly.com/openbook/linuxdrive3/book/>
- c) http://en.wikipedia.org/wiki/Comparison_of_lightweight_web_servers
- d) <http://perldoc.perl.org/CGI.htm>