

## Trabajo práctico III

# Sistemas Embebidos

---

### Introducción

En la intersección de Software, Hardware y Comunicaciones nos podemos encontrar a los Sistemas Embebidos. Los mismos son sistemas que, si bien su definición varía con la literatura, se pueden definir como computadoras de uso específico, es decir, computadoras con requerimientos de hardware, software y comunicaciones bien definidos.

Es por esto la importancia que poseen este tipo de sistemas para los Ingenieros en Computación, y que da origen al presente práctico.

### Objetivo

El objetivo del presente trabajo práctico es que el estudiante sea capaz diseñar una aplicación para un sistema embebido.

### Desarrollo

Se pide que sobre un sistema embebido tipo Raspberry Pi o similar, que posea MMU, se desarrolle una aplicación para un sistema embebido, que debe cumplir con:

- 1) Instalar un sistema operativo GNU/Linux en el sistema embebido.
- 2) Realizar 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.

- 3) 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) Página que reporte información sobre recursos varios del sistema embebido:
    - i) Procesador
    - ii) Memoria
    - iii) Uptime
    - iv) Fecha y Hora actual
  - b) Página que permita enviarle comandos al *baash*, utilizando en el TP1, y los muestre en pantalla (simil a una terminal html). NO implementa connect, descargar 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.

## Entrega

- 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), y explicando cada script y programa implementado.
- 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).



## Evaluación

El presente trabajo práctico es individual deberá entregarse antes de las **23:55 (UTC -3)** del día **domingo 3 de junio de 2018** mediante el LEV. Será corregido y luego deberá coordinar una fecha para la defensa oral del mismo.

## Referencias y ayuda

[1] <http://www.linuxdevcenter.com/pub/a/linux/2007/07/05/devhelloworld-a-simple-introduction-to-device-drivers-under-linux.html?page=1>

[2] <http://www.oreilly.com/openbook/linuxdrive3/book/>

[3] [http://en.wikipedia.org/wiki/Comparison\\_of\\_lightweight\\_web\\_servers](http://en.wikipedia.org/wiki/Comparison_of_lightweight_web_servers)

[4] <http://perldoc.perl.org/CGI.htm>

