Información General

Conocimientos Previos:

Experiencia en programación en C, conocimientos básicos de sistemas operativos, usuario Linux/Unix en línea de comando, editores (vi, emacs).

Horario:

Curso Drivers: Lunes 5, martes 6 y Miércoles 7 de marzo de 2018 de 9 a 18 hs.

Matrícula:

Curso Drivers en Linux: 1280 €
Descuento asistente Linux empotrados (15%)*: 1088 €.
Descuento asistente otros cursos (10%): 1152€.

El precio incluye el material del curso, comidas y cafés.

*Si asiste además al curso Linux empotrados ó Linux tiempo real se llevan el sistema de desarrollo.

Forma de pago:

Una vez recibida la confirmación de la reserva de plaza, se le enviará una hoja de inscripción para pago por tarjetas de crédito ó transferencia Bancaria. Información adicional sobre facturación y pagos: facturacion@electratraining.org

Lugar:

Escuela Politécnica Superior de la Universidad Autónoma de Madrid, Ctera. de Colmenar Km.15, 28049 Madrid. Ver como llegar a la EPS - UAM en nuestra web. La Escuela se encuentra comunicada por autovía (M-40 + M-607 Ctra. de Colmenar), autobuses y trenes de cercanías. No existen problemas de aparcamiento.

Reserva de Plaza:

El curso tiene un cupo limitado a 16 asistentes. Puede reservar plaza vía la página web del curso (www.electratraining.org) o enviando los siguientes datos por correo electrónico.

Nombre y Apellido:
Empresa:
Teléfono de contacto:
e-mail:
Fax:
Reserva de Plaza:
Curso Drivers Linux 2018

Información Adicional:

Gustavo Sutter

Ctra. de Colmenar Km.15
28049 Madrid

Teléfonos: 91 497 6209/ 609 691 607
Correo: gustavo.sutter@uam.es
Internet: http://www.electratraining.org/

Desarrollo de Sistemas Linux Empotrado: Drivers

Ciclo 2018 de Entrenamiento Intensivo en Diseño de Sistemas

Lugar de realización

Escuela Politécnica Superior

Universidad Autónoma de Madrid

Cursos 2018: Desarrollo de Drivers en Linux

Presentación

El curso se dirige a las empresas interesadas en trabajar con Linux para la fabricación de sus productos, ya sean del sector industrial, telecomunicaciones o electrónica de consumo. La tendencia del mercado se dirige a la incorporación de valor añadido a las funcionalidades básicas de los productos utilizando para ello código contrastado, en contra del desarrollo propietario de cada nueva funcionalidad.

El objetivo del curso es conocer el funcionamiento del kernel de Linux y la interfaz de programación interna del kernel. Durante el curso se introducirá progresivamente el API de programación del kernel, con ejemplos de programación de drivers utilizando módulos dinámicos. Se implementarán varios drivers para control de sensores añadidos a las placas utilizadas para el curso, BeagleBone (ARM Cortex A8). Además se utilizarán herramientas de emulación/virtualización para facilitar y acelerar las fases de desarrollo. También se estudiaran casos prácticos planteados por los alumnos.

El curso tiene un marcado carácter práctico introduciendo los conceptos teóricos necesarios para su ejecución. La duración del curso es de 24 horas distribuidas a lo largo de 3 días de trabajo en dos sesiones de mañana y tarde.

Herramientas Disponibles

Cada puesto de trabajo dispondrá de un ordenador configurado con el sistema Linux, placa de desarrollo empotrada basada en ARM Cortex A8, manuales teórico-práctico y biblioteca de software con ejercicios prácticos.

Prerrequisitos

Conocimientos avanzados de sistemas operativos y de usuario Linux/Unix. Línea de comandos shell Linux/Unix. Editor de texto (vi, emacs). Programación en lenguaje C.

Este curso se relaciona con los cursos **Desarrollo de Sistemas Linux Empotrados** (mayo 2018) y **Programación en Linux Tiempo Real** (octubre 2018). Existen descuento por asistencia a más de un curso (*). Más información en

http://www.electratraining.org

* además asistiendo a dos de los tres cursos se lleva la placa de desarrollo

Descriptores Curso Desarrollo de Drivers en Linux

1. Introducción al sistema Linux

   GNU Linux. Diseño y Arquitectura.

2. El kernel de Linux. API de programación

   Módulos dinámicos: creación y compilación. Gestión de dispositivos: estructuras, interfaz de programación. Gestión de la memoria: kmalloc and cia, DMA, caches, mmap. Gestión de interrupciones: registro, tratamiento, tiempo real. Planificación de procesos: kernel threads. Temporización: timer del sistema, kernel timers. Sincronización: atomicidad, semáforos, barriers, spinlocks, mutex

3. Depuración del kernel

   La función printk, interfaz /proc, GDB & KGDB & LKCD, debugfs, kprobes, qemu.

4. Discusión de Casos Reales