



Expte.Nº 0023112/2009

RESOLUCION HCD N° 193/2009.-

VISTO:

La presentación efectuada por los Lics. Nicolás Wolovick y Daniel Moisset, solicitando la aprobación del Curso de Extensión “Construcción de un driver de tarjeta WiFi para el núcleo de Linux”, dirigido a público en general con formación en Sistemas Operativos y Redes de computadoras;

CONSIDERANDO:

Que acompañan el Programa del curso propuesto y los detalles de su implementación;

Que en su dictamen la Comisión de Extensión de este Cuerpo aconseja dar curso favorable a la solicitud de los Lics. Wolovick y Moisset;

POR ELLO

EL HONORABLE CONSEJO DIRECTIVO DE LA  
FACULTAD DE MATEMÁTICA, ASTRONOMÍA Y FÍSICA  
RESUELVE:

ARTÍCULO 1º: Aprobar el dictado del Curso de Extensión “Construcción de un driver de tarjeta WiFi para el núcleo de Linux”, destinado a público en general con formación en Sistemas Operativos y Redes de computadoras, según lo especificado en el Anexo que se acompaña formando parte de la presente.


ARTÍCULO 2º: Autorizar a los Lics. Wolovick y Moisset, profesores de esta Institución, a dictar el curso mencionado precedentemente, sin perjuicio de la actividad académica que desarrollan en Fa.M.A.F.

ARTÍCULO 3º: Comuníquese y archívese.

DADA EN LA SALA DE SESIONES DEL H. CONSEJO DIRECTIVO DE LA FACULTAD DE MATEMÁTICA, ASTRONOMÍA Y FÍSICA A VEINTISIETE DÍAS DEL MES DE JULIO DE DOS MIL NUEVE.-

gl

  
Dr. WALTER N. DAL LAGO  
Secretario General Fa. M. A. F.

  
Dr. DANIEL E. BARRACO DÍAZ  
DECANO  
Fa.M.A.F.



Anexo Res. HCD N° 193/2009

**Docentes responsables**

Lic. Nicolás Wolovick, Lic. Daniel Moisset

**Docente que dictará el curso**

Christoph Hellwig

**Título del curso**

“*Construcción de un driver de tarjeta WiFi para el núcleo de Linux*”

**Objetivos generales**

- Formar a un grupo de personas en el estudio, desarrollo y mantenimiento de drivers para el núcleo de Linux.
- Obtener un producto de software concreto y hasta el momento no disponible en el núcleo de Linux.
- Atender la demanda de una parte de la sociedad, específicamente la industria informática, que cada vez más requiere programadores con conocimientos profundos de tecnologías usadas ampliamente en el mercado.

**Objetivos específicos**

Contribuir al universo FLOSS (Software Libre y de Código Abierto) con un driver aún inexistente dentro del segmento de placas de red inalámbricas IEEE 802.11 con interfase USB, ampliando la base de hardware compatible con el sistema operativo GNU/Linux en base a un desarrollo colaborativo y de fuerte base regional.

Entrenar a un grupo de personas en todo el proceso de desarrollo colaborativo de un driver para una pieza específica de hardware, empezando desde el estudio del hardware mediante sus hojas de datos, hasta la obtención del módulo de kernel que brinde todas las funcionalidades de red esperadas.

**Destinatarios y cupo de alumnos:**

Público en general, con formación comprobable (no necesariamente formal) en Sistemas Operativos y Redes de Computadoras.

El curso será en *inglés* por lo que se requiere un manejo fluido de ese idioma.

El cupo será de 12 personas y la limitación está dada principalmente por la cantidad de personas que el docente puede manejar y el número de placas WiFi que dona VIA Technologies, Inc. a través de Hellwig.

Se priorizará aquellos interesados con habilidades para el desarrollo colaborativo, con buen manejo del lenguaje “C”, así como todo el conjunto de herramientas de desarrollo GNU. También se priorizarán aquellas personas que hayan tenido experiencia desarrollando dentro de algún núcleo de sistema operativo.

La selección se realizará en base al CV que los postulantes presenten. Se formará un comité ad-hoc de selección de postulantes y se propone la siguiente conformación:

- Lic. Nicolás Wolovick (docente de Fa.M.A.F – U.N.C.),
- Lic. Daniel Moisset (docente de Fa.M.A.F – U.N.C., miembro de GrULiC),
- Federico Heinz (presidente de la Fundación Via Libre, estudiante de Fa.M.A.F. - U.N.C.).

**Lugar en que se dictará el curso**

Sala de Computación, 3er piso, Fa.M.A.F., Universidad Nacional de Córdoba.

**Duración, carga horaria y fechas estipuladas de las clases:**

Todos los martes y jueves de Agosto de 17 a 20 horas. (desde el martes 4 de Agosto al jueves 27 de Agosto).

Serán 24 horas de clase presencial divididas en 8 clases de 3 horas cada una, y se espera una carga similar (24 horas) de trabajo no dirigido en la casa, totalizando un curso de **48 horas**.





Cada clase consistirán en una introducción de 30 minutos a cargo del docente al tema relevante del día, para luego empezar a desarrollar de manera supervisada por los docentes.

### **Contenidos**

Se seguirá principalmente los contenidos propuestos en el libro "*Linux Device Drivers*", 3ra edición.

**Introducción a los Device Drivers.** Su rol, dividiendo el kernel, clases de dispositivos y módulos, problemas de seguridad, versiones, licencias, como desarrollar junto a la comunidad del Kernel de Linux.

**Construyendo y corriendo Módulos.** Preparando el sistema, «Hola Mundo», compilando y cargando, la tabla de símbolos, inicialización y descarga, parámetros de entrada.

**Dispositivos de Caracteres.** Números mayor y menor, estructuras de datos importantes, registro de dispositivos de caracteres, open-read-write-release.

**Técnicas de Debugging.** Soporte de debugging en el kernel, imprimiendo, preguntando, midiendo, fallos de segmento, debuggers y otras herramientas.

**Concurrencia y Condiciones de Carrera.** Problemas de la concurrencia, manejo de la concurrencia, semáforos y mutex, spinlocks, locking traps.

**Manejo avanzado de Dispositivos de Caracteres.** Ioctl, E/S bloqueante, poll() y select(), notificación asíncrona, buscando un dispositivo, control de acceso de un dispositivo.

**Tiempos, Esperas y Ejecución Diferida:** midiendo lapsos de tiempo, obteniendo la hora, ejecución diferida, temporizadores del kernel, tasklets, workqueues.

**Manejo de Memoria.** Kmalloc, lookaside caches, vmalloc, variables-por-cpu, como obtener buffers grandes.

**Comunicación con el Hardware.** E/S mapeada en puertos y en memoria, ejemplos, usos.

**Manejo de Interrupciones.** Ejemplo con el puerto paralelo, instalando un manejador de interrupciones, implementación del manejador, mitad superior e inferior, compartir interrupciones, E/S manejada por interrupciones.

**Tipos de Datos del Kernel.** Uso de los tipos de dato estándar de "C", como asignar tamaños explícitos a los tipos, tipos específicos para las interfaces, problemas de portabilidad, listas ligadas.

**Drivers USB.** Conceptos básicos, SysFs, USB Request Blocks (URB), como escribir un driver USB, transferencia sin utilizar URBs.

**El Modelo de Dispositivos del Kernel.** Kobjects, ksets, operaciones de bajo nivel de SysFs, generación de eventos hotplug, buses, dispositivos y drivers, clases, hotplugging, trabajando con firmware.

**Drivers de Red.** Diseño de snull un ejemplo de driver de red, conectandose al kernel, la estructura net\_device, abriendo y cerrando, transmisión y recepción de paquetes, manejo de interrupciones, recepción de migración de interrupciones, cambios en el link-state, buffers para sockets, resolución de direcciones MAC, comandos ioctl especiales, información estadística, multicast.

### **Bibliografía**

□ Jonathan Corbet , Alessandro Rubini, Greg Kroah-Hartman, "*Linux Device Drivers*", 3ra edición, Febrero 2005, O'Reilly.

□ Christian Benvenuti , "*Understanding Linux Network Internals*", Diciembre 2005, O'Reilly.

### **Requisitos de Aprobación:**

Entregar el driver desarrollado con la funcionalidad mínima que permita utilizar la placa WiFi USB VIA como interface para de red en una sesión de trabajo (conexión a redes WiFi con y sin cifrado, autoconfiguración por DHCP, sincronización NTP, consulta de correo, transferencia de archivos y navegación Web).

### **Modalidad**



Presencial.

**Equipamiento necesario para el dictado.**

Sala de Computación (esto ya fue mencionado a los integrantes de Computación y no existen superposiciones ni restricciones).

Permitir la entrada y salida de los estudiantes y docentes en el horario definido (tener en cuenta que la clases terminan a las 21:00).

En el llamado se explicitará que los alumnos deben contar con una computadora personal portátil que permita trabajar en el desarrollo. En caso de existir candidatos con las condiciones requeridas pero que no posean su notebook, la Facultad podrá decidir apoyarlos, dependiendo de la disponibilidad de equipos, facilitando durante el mes que dura el curso dos o tres PCs de escritorio que sirvan para desarrollar. Para su uso dichas PCs deberán ser reinstaladas con un Linux conteniendo el último kernel disponible (2.6.30). **La falta de PCs de escritorio para prestar a los alumnos por parte de la Facultad, no es una condición que impida la realización del curso.**

Acceso a las redes inalámbrica para realizar las pruebas de los drivers, en particular al Access Point “ap2.cs” de Computación que está en la misma Sala de Computación. Los otros Access Points públicos (FaMAF1-5) se utilizarán para pruebas de alcanzabilidad y comportamiento frente a señales más débiles.

**Factibilidad económica (arancel estipulado, en caso que corresponda, y destino de los fondos)**

El curso será gratuito, además el docente no cobrará honorarios. La empresa VIA Technologies, donará a la Facultad 15 placas WiFi USB para desarrollar y mantener el driver. Dichas placas ya han sido enviadas al Lic. Wolovick por correo a su dirección postal de Fa.M.A.F. .

**Otra información**

El curso finalizará con la obtención de un driver para el kernel de Linux que aun no existe, por lo que el grupo de personas a cargo de la experiencia se convertirá en *maintainers* de dicho módulo, continuando su contribución al mundo FLOSS.

Luego de esta experiencia será posible generar Proyectos de Extensión relacionados con esta temática específica, el desarrollo abierto y colaborativo de drivers para Linux.