



UXI



REVISTA DE SOFTWARE LIBRE DE LA UCI

FEBRERO / 2008

“Para un festival tengo diez razones”

Noticias

“Lanzamiento de la Comunidad Virtual Unix-BSD de la UCI”

Migración

“Herramientas para el acceso a los los servicios de Internet”

Soluciones de esta edición

“Grub en un disquete”
“Cómo crear un GrubSplash”

Programación

“Programación de hilos y semáforos en GNU/Linux con C++”

Entrevista ¿X?

“Comunidad de Software Libre por la universalización del conocimiento”

Humor Libre

Seguridad y Redes

“Apache como servidor web por excelencia”

Eventos

“Festival de Software Libre hasta la Lenin”

Informática 2007

“Open Source Business Intelligence”

“Proceso de Desarrollo de Aplicaciones de acuerdo a los conceptos de Software Libre”



1er Aniversario

Publicar nuestro software: único camino hacia la libertad plena.

10 números que no caben en 100

“En estos momentos, **UXi** se encuentra referenciada en diversos sitios del mundo del software libre, como Ubuntu.es y Barrapunto.com”



...Página 1



grub>

GRUB en un disquete

...Página 11

Cómo construir un BootSplash

...Página 12





Jefe Consejo Editorial:



Abel García Vitier
avitier@estudiantes.uci.cu

Editores:



Jorge Luis Betancourt González
jlbetancourt@estudiantes.uci.cu



Evelio Clavel Rosales
eclavel@estudiantes.uci.cu



Yosbel Brooks Chávez
ybrooks@estudiantes.uci.cu

Redacción:



Yailin Simón Mir
ymir@estudiantes.uci.cu



Dunia Virgen Cruz Góngora
dvruz@estudiantes.uci.cu



Daina M. Gonzáles Matos
dmgonzales@estudiantes.uci.cu

Coordinadores:



Ing. Abel Meneses Abad
abelma@uci.cu



Eiger Mora Moredo
emora@estudiantes.uci.cu

Arte y Diseño:



Angel Alberto Bello Caballero
aabello@estudiantes.uci.cu



David Padrón Álvarez
dpadron@estudiantes.uci.cu



Ezequiel Manresa Santana
emanresa@estudiantes.uci.cu

Revisión y Corrección:



MSc. Clara Gisela Scot Bigñot
claragisela@uci.cu



MSc. Graciela González Pérez
gracielagp@uci.cu



Rislaidy Pérez Ramos
rpramos@estudiantes.uci.cu

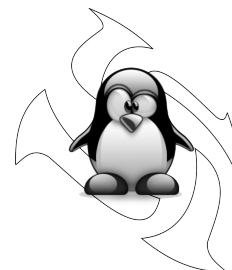


Victor Frank Molina López
vfmolina@estudiantes.uci.cu



Félix Daniel Batista Diñeiro
fdbatista@estudiantes.uci.cu

Nuestros Amigos:



Movimiento de Software Libre
Facultad 2



Revistas digitales cubanas de Informática



Grupo de Producción FEU
Facultad X



Servicios Especializados
para la Migración a SWL





Editorial

Estimado Lector:

UXi llega a usted una vez más, con esta edición que inicia el segundo volumen de publicaciones, un volumen que constará de diez ediciones en la que esperamos cumplir con nuestro objetivo de difundir el Software Libre en nuestra Universidad y en el mundo.

Estamos muy complacidos con el Festival de Software Libre realizado el 31 de enero dedicado a nuestro Primer Aniversario y por ello abrimos UXi 01 Vol. 02 con un artículo dedicado a este evento.

Podrá disfrutar de las habituales secciones de UXi. Proponemos a nuestros lectores dos interesantes soluciones "Grub en un disquete" y "Cómo construir un GrubSplash"; además de un artículo en la sección Programación dedicado a la "Programación de Hilos y Semáforos en Linux con C++". Retomamos la sección Entrevista ¿X?, esta vez con una entrevista realizada al líder de la Comunidad de Software Libre de la Universidad de las Ciencias Informáticas (UCi) y propósito de las actividades de dicha comunidad, algunos miembros entre los que se incluye nuestro Grupo Editorial, hemos estado realizando festivales de Software Libre en algunos centros educacionales de la capital y hemos dedicado la sección Eventos al Festival desarrollado en el IPVCE Vladimir Ilich Lenin.

En fin, esperamos que este número sea de su agrado y que constituya una muestra de nuestro esfuerzo por llegar a ser una publicación oficial. Desde ya lo invitamos a leerla y comentarnos sus críticas y sugerencias a través de la dirección de correo softwarelibre@uci.cu, a través de la cual también puede suscribirse a UXi si pone en el asunto "Suscribir UXi". Invitamos a todos nuestros lectores a enviarnos artículos que deseen publicar.

Atentamente,
Grupo Editorial

Se otorga permiso para copiar, distribuir y/o modificar este documento bajo los términos de la Licencia de Documentación Libre de GNU v1.2 o cualquier otra versión posterior publicada por la FSF; sin Secciones Invariantes, Textos de Cubierta Delantera ni Trasera. Puede consultar una copia de la licencia en:

<http://www.gnu.org/copyleft/fdl.html>





Para un festival tengo diez razones

Lo prometido es deuda, UXi abre este número con un artículo dedicado a la celebración de su Primer Aniversario que devino en Festival de Software Libre y demostró que cuando se quiere, se puede...



Noticias

En esta sección se ha incluido el Lanzamiento de la Comunidad Virtual Unix-BSD de la UCI...



Migración

Se presenta un estudio de todas las herramientas libres encontradas en la red para el acceso a los servicios de internet, perteneciente a una serie de artículos y estudios que irán saliendo en cada número de UXi...



Soluciones de esta edición

En esta ocasión se presentan dos soluciones, la primera con 10 comandos para tener el GRUB en un disquete, y la segunda nos enseñará a construir un GrubSplash utilizando GIMP para "ponerle" al Grub la cara que se desee...



Programación

En este artículo se abordan los aspectos más importantes de la programación de hilos y semáforos, utilizando Code::Blocks...



Entrevista ¿X?

Una entrevista a Marcos Ortiz Valmaseda, coordinador de la Comunidad de Software Libre de la UCI, dará la oportunidad de conocer más sobre sus temas polémicos y los resultados alcanzados...



Humor Libre



Seguridad y Redes

En la edición UXi 9 en esta misma sección y con el tema "Servicios telemáticos sobre GNU/Linux" se anunciaba una serie de artículos relacionados con este tópico. Sin intención de establecer un orden jerárquico comenzaremos por el servidor Web Apache...



Eventos

El primer encuentro con los estudiantes de la Lenin, nos dejó bastante consternados. No parecían ni por mucho interesados en el Software Libre, pero la realidad demostró la validez de aquel refrán popular: "Ver para creer"...



Informática 2007

Una nueva edición de UXi, acompañada de dos resúmenes más de trabajos presentados en el evento Informática 2007 ...

... Página 1

... Página 2

... Página 6

... Página 11

... Página 14

... Página 19

... Página 20 y
24

... Página 21

... Página 25

... Página 27



/etc/init.d/uxi start

Para un festival tengo diez razones

Felix Daniel Batista
fdbatista@estudiantes.uci.cu
Movimiento "Código y Letra"



El Software Libre estuvo de plácemes el pasado jueves 31 de enero en el tercer piso del Docente 3 de la Universidad de las Ciencias Informáticas (UCi), lugar devenido escenario del Festival de Software Libre, un tradicional evento que en esta ocasión fue celebrado con motivo del primer aniversario de la salida a la luz de la revista UXi.

La buena organización que predominó en todos los puestos y áreas expositivas fue resultado de los esfuerzos conjuntos de la dirección de la facultad 10, del proyecto Unicornios y del consejo editorial de la propia revista –apéndice importante del citado proyecto–, quienes, contando únicamente con sus limitados recursos, realizaron un enorme esfuerzo para montar la excelente exposición que finalmente lograron.

Los allí presentes pudieron apreciar de primera mano los logros cosechados por la comunidad linuxera de la UCI. Entre ellos, mención especial merece la popular distribución Nova, desarrollada por cerca de 30 estudiantes de la facultad 10, y que se encuentra inmersa en un activo proceso de mejora y mantenimiento.

De forma paralela a la muestra expositiva de productos informáticos, se impartieron diferentes conferencias sobre los más diversos y novedosos temas. Por ejemplo, Victoria Isabel Pérez, especialista de la Empresa de Consultoría y Seguridad Informática SEGURMATICA, hizo referencia a las diferentes variantes de programas malignos que corren sobre el sistema operativo GNU/Linux.

También se abordaron otros temas sumamente sensibles, como el impacto del software libre en la educación superior y las experiencias de trabajo con metodologías ágiles de desarrollo.



UXi: diez números que no caben en cien

La razón de ser de este festival, sin embargo, fue la celebración del aniversario de la revista de software libre de la UCI, bautizada UXi por sus creadores hace exactamente un año.

Según palabras de Abel García, quien está al frente del consejo editorial, al principio del proyecto ninguno de los tres miembros iniciales pudo prever la magnitud que alcanzaría la publicación a solo 12 meses de haber sido fundada.

“En estos momentos, UXi se encuentra referenciada en diversos sitios del mundo del software libre, como Ubuntu.es y Barrapunto.com”, comentó Abel Meneses, al frente de la comisión organizadora del festival y líder del proyecto Unicornios. “Contamos con la presencia en el grupo editorial de estudiantes de cinco facultades de la UCI y la regional de Granma, lo cual aporta variedad y representatividad a los números, además de que contribuye a llevar el espíritu de la UCI a las facultades regionales y viceversa”.

Desde UXi-01 –primer número que vio la luz hasta UXi-X, la décima edición correspondiente a enero de este año, el objetivo de la revista siempre ha sido el mismo de los festivales y exposiciones de software libre que se han realizado: apostar por el franco acceso a la información y la socialización del conocimiento científico, armas que debemos esgrimir siempre en nuestra batalla por la soberanía tecnológica.



Hora de la transición GNU/Linux

El próximo 30 de junio Microsoft planea dejar de distribuir Windows XP en su versión OEM, lo que implica que los vendedores de equipos nuevos solo podrán entregarlos pre instalados con Vista. El service pack 1 de Vista, estará disponible al consumidor a partir de marzo. Quizás sea hora de considerar la opción de hacer una transición.

GNU/Linux ha tenido un vertiginoso desarrollo en las distribuciones, las mismas ofrecen un sistema operativo estable, ligero y seguro, una suite de oficina competitiva y abierta, no son propensos a virus, y sobre todo, no requieren hardware de última generación para instalarse y funcionar. Además, estas distribuciones representan ahorros considerables contra el licenciamiento de Windows - Office, que van desde un 50 hasta un 100%.

<http://softwarelibre.uci.cu>

GNU/Linux con Photoshop

Dan Kegel, representante de Google, informó que la compañía había pagado a CodeWeavers, que también administra el desarrollo de Wine, por hacer posible ejecutar Photoshop de manera estable en GNU/Linux.

El resultado de la labor encomendada a los desarrolladores fue aproximadamente 200 parches que han sido implementados en la versión 0.9.54 de Wine. La compañía Adobe también ha registrado la información, aunque precisa que por ahora no tiene planes de lanzar una versión verdadera de Photoshop para GNU/Linux.

<http://softwarelibre.uci.cu>

Las dos Koreas se unen por GNU/Linux

Korea de Norte y Korea del Sur colaborarán para crear una distribución de GNU/Linux llamada "Hana Linux" y un conjunto de estándares. En un foro llevado a cabo recientemente en China, expertos de IT de ambas Koreas se pusieron de acuerdo en la necesidad del desarrollo en el sector de GNU/Linux, además de otros proyectos como OpenOffice.org y soporte del lenguaje coreano para software Open Source.

<http://softwarelibre.uci.cu>

El Software Libre gana terreno en Cuba

David Grau Merconchini, especialista del Centro de Información y Gestión Tecnológica de Santiago de Cuba, asegura que la introducción paulatina de software libre en Cuba resulta muy importante para avanzar en la informatización de la sociedad. Precisa que en 1990 se comenzaron a usar las tecnologías libres en Cuba, en un proceso no del todo convencional. Actualmente hay más de 2 000 usuarios de distribuciones GNU/Linux. Existe un grupo multidisciplinario, encabezado por la Oficina Nacional de Informatización, encargado de analizar las diferentes experiencias de empleo de software libre, además de directivas para su implementación paulatina.

Fuentes: RHC, AIN, GRANMA, TRABAJADORES, JUVENTUD REBELDE, PL, REUTER, EFE, IPS, ANSA, AFP, XINHUA, TASS, DPA, AP.

Vacuna contra Vmsplice en la UCI

Ha salido en internet la vulnerabilidad en el kernel relacionada con vmsplice, que afecta el kernel en las ramas del 2.6.18 en adelante. El problema, descubierto por Wojciech Purczynski indica que el fallo permite un acceso como root a una máquina con GNU/Linux aprovechando que se conoce la teórica posición en memoria de la función vm_splice. Al realizarse una llamada no autenticada, se produce un fallo que permite hacerse con el control de la máquina.

El grupo de servicios especializados para la migración a software libre en la UCI pone disponible para toda la comunidad el parche del kernel y el kernel 2.6.18-X86 ya parcheado para quienes deseen instalarlo libremente. Esta solución esta disponible en el Servidor [Documentación](#).

<http://softwarelibre.uci.cu>

La Comunidad de SWL en Artemisa se crece

Facultad Regional de Artemisa crea su propio portal de Software Libre. Su coordinador Rolando Santamaría Masó le escribe a la comunidad de software libre de la UCI para informarles de sus logros con el desarrollo de su propio portal.



La comunidad UCI de Software Libre de la UCI ha desarrollado este año un grupo de actividades sobre la red: Festivales de software libre UCI, Lenin y próximamente Humbolt; también han comenzado el despliegue de soluciones de software importantes para la comunidad nacional: Nova, Infodrez, Portal de SWL 2, Evolution Config Exchange, Ecumenix; una recontextualización de los servicios como el repo Debian Alternativo con los paquetes: NetBeans, Diccionarios del Stardict, ZendStudio, Nvu, entre otros.

<http://softwarelibre.uci.cu>

Entrevista a un desarrollador de XFCE

En una entrevista a Benedikt Meurer, a propósito del apreciable ascenso del entorno de escritorio tipo Unix, destacó que una de sus ventajas sobre KDE y GNOME es su simplicidad. Uno todavía puede conocer el código base en menos de una semana y ser capaz de entender las decisiones de su diseño básico. De esta manera XFCE 4.x todavía tiene mucho potencial, mientras que otros necesitan ser reescritos (KDE ya comenzó una reescritura y muchos de los mantenedores de GNOME están votando por una reescritura para su v3.0).

Artículo completo en Lxer.
Fuente Español: VivaLinux.com.ar

Se difunden conocimientos de Blender en la UCI

A fin de difundir el conocimiento sobre Blender y sus posibilidades, la Comunidad Universitaria desarrolla Talleres Universitarios.

Esta herramienta libre, exponente representativa de la creación de contenidos tridimensionales, cuenta además con una comunidad activa a nivel internacional que le ha merecido el reconocimiento de diseñadores y artistas de la plástica.

Es por esta razón que nuestra Universidad, inmersa en un proceso paulatino de migración tecnológica tiene a Blender como una alternativa ideal.



<http://softwarelibre.uci.cu>

En Francia, la Gendarmería migrará sus sistemas a GNU/Linux

Francia, la Gendarmería ha anunciado que migrará sus sistemas informáticos a Ubuntu y prescindirá de los productos de Microsoft, informa Associated Press. El cambio a Ubuntu de los 70.0000 ordenadores con los que cuenta se realizará progresivamente, de manera que todos los ordenadores funcionarán con el sistema libre antes de 2014. Sus razones se resumen en no querer depender sólo de una multinacional, tener control absoluto sobre su herramienta informática, y su menor coste. La Gendarmería se convierte en una de las administraciones más grandes en migrar a sistemas de código abierto, aunque no son los primeros en Francia.

[Fuente](#)

Megadictos

La revista digital cubana Megadictos ha demostrado en poco tiempo lo que puede hacerse cuando hay voluntad y empeño. Los integrantes de este proyecto, quienes apoyan el movimiento de Software Libre y la gestión del conocimiento convocan a todos los interesados en colaborar con Megadictos a que los contacten a través de la dirección electrónica megadictos@gmail.com y/o también suscribirse a su lista de discusión megadictos@rbol.org. Usted puede encontrar las publicaciones de estos amigos y colaboradores de UXi en el Servidor de [Documentación](#).

Filipinas migra a GNU/Linux

La alternativa de código abierto tomó fuerza en Filipinas a raíz de la crisis asiática de 1997. La migración a GNU/Linux no fue fácil debido a las dudas de las autoridades y a la implantación y subvenciones de Microsoft que ofrecía XP a 20 dólares y la suite ofimática Office a 30 dólares. Ricardo González, asesor independiente de software libre de Filipinas, declaró que aún así les había salido más barato todavía. GNU/Linux y el software libre son la mejor forma de dotar de recursos a la educación.

En diciembre de 2007, se enviaron 10 mil equipos con dos distribuciones: Kubuntu y Edubuntu. El ahorro realizado permitió que el gobierno de Manila aportara 3 mil computadores adicionales que beneficiarán a 300 colegios.

[Fuente](#)



Gnome Simple Stateful Music Player 1.2

Gnome Simple Stateful Music Player (GSSMP) es un reproductor de música diseñado para trabajar bajo Gnome. Al contrario de la mayoría de los reproductores no almacena tu colección de audio en una base de datos y sólo los organiza por archivos y carpetas. Es realmente muy pequeño y prácticamente no consume recursos. Bien a lo Gnome, hace lo que tiene que hacer sin hacer preguntas. Una buena opción para aquellos que no poseen demasiados recursos o que solo quieren escuchar algunos temas de vez en cuando.

<http://softwarelibre.uci.cu>

Castigan a un alumno por usar Firefox

Un estudiante de secundaria de Pennsylvania ha sido amonestado por tener el atrevimiento de ejecutar Firefox en clase. El incidente comenzó con la advertencia del profesor sobre la utilización de esa herejía a lo que el alumno le indicó que era un navegador web alternativo y que estaba trabajando con él. No sirvió de nada la explicación y tampoco cuando el alumno insistió que era el mejor navegador y que no estaba haciendo nada incorrecto. Tras varios avisos de su "educador", fue castigado con dos horas de arresto y otras cuantas en fin de semana para purgar sus pecados o en su defecto tres días de expulsión del colegio.

<http://softwarelibre.uci.cu>

Samsung comienza la producción de los SSDs de 64 GB SATA II

La multinacional coreana ha comunicado el inicio de la producción "en masa" de unidades de estado sólido de 64 Gbytes de capacidad e interfaz SATA II, con los que debe comenzar el esperado descenso de precio de esta tecnología. Las nuevas unidades serán un 60% más rápida que las equipadas con SATA I alcanzando transferencia de datos de 100 MB/s en velocidad de lectura y 80 MB/s en escritura. Estarán disponibles próximamente como opción en portátiles Dell y Alienware, según Samsung.

Fuentes:
SoftwareLibre.Net
fentlinux.com

Bos Wars 2.4.1



Ha salido la nueva versión disponible de **Bos Wars**, un juego para los amantes de la estrategia en tiempo real, que hace uso del motor Stratagus y se asemeja en el sistema de juego al clásico Warcraft II. En esta entrega **2.4.1**, se ha corregido un error en el guardado y carga de partidas, se ha incluido un nuevo sistema para definir mapas y el modo OpenGL es ahora seleccionable desde las opciones.

Fuente: fentlinux.com

Versión 0.9.56 del emulador

Se encuentra disponible una versión del emulador de programas de *Windows en Gnu/Linux*, *Wine*, en concreto la versión **0.9.56**. Mediante esta aplicación podremos hacer correr programas del sistema operativo Windows en nuestro adorado GNU/Linux.

BlackHat "Un proyecto para todos"

BlackHat es un proyecto de divulgación científico-técnica para todos los interesados en la Informática y las ciencias afines, implementado con el objetivo de elevar la cultura informática en Cuba mediante la creación de una comunidad de profesionales y estudiantes involucrados en el tema.

Esta Comunidad cuenta con un boletín digital que distribuyen a través del correo a sus suscriptores. La misma cumplió recientemente su primer año de existencia con 44 boletines publicados. Desde UXi le enviamos una felicitación a estos amigos.

Usted puede colaborar y ser miembro de esta comunidad, compartiendo información, ya sean artículos, códigos, o cualquier otro asunto que resulte útil a la comunidad, además recibir el boletín digital vía e-mail. Para ello escriba a blackhat4all@gmail.com.

Desde la UCI puede acceder a todas las publicaciones de BlackHat disponibles en el servidor de [Documentación](#).

Visite y Suscríbese a [BlackHat](#) y no se arrepentirá.





Lanzamiento de la Comunidad Virtual Unix-BSD de la UCI

Marlon O. Figueredo La Cruz
mfigueredo@estudiantes.uci.cu
Facultad X



En el marco de la celebración por el 1er Aniversario de UXi, la Revista de Software Libre, el pasado 31 de enero, se lanzó la Comunidad Virtual Unix-BSD de la UCI con el nombre "Virtual Community Unix-BSD". Usted puede ser parte de esta comunidad en la UCI, sobre los Sistemas Operativos BSD, sólo debe confirmarlo voluntariamente mediante el envío de un e-mail a mfigueredo@estudiantes.uci.cu o contacto personal con Marlon O. Figueredo La Cruz, será parte del Grupo de Usuarios BSD en la UCI(UciGUB) o The BSD User Group UCI (UciBUG)).

¿Qué es la Comunidad Virtual Unix-BSD de la UCI?

Es la primera de su tipo, en la cual los integrantes interactúan para satisfacer sus necesidades y comparten un propósito determinado que constituye la razón de ser de la comunidad virtual. Además tiene una política que guía las relaciones y cuenta con los sistemas informáticos que median las interacciones y facilitan la cohesión entre los miembros.

La actividad diaria de la comunidad será peculiar, exclusiva y novedosa pues estará completamente soportada sobre TCP/IP o Internet, la tecnología nativa de los BSD, específicamente en listas de distribución como el primero de los canales de comunicación y en el servicio de mensajería instantánea.

Se prevé la creación de un sitio Web con acceso a las páginas internacionales de los BSD, tendrá servicio de documentación, foro de discusión, una sección de FAQ, noticias propias de la comunidad universitaria así como una página de descarga de todos los ISOs BSD.

La comunidad se definirá por 2 rasgos distintos:

1. La Comunidad Virtual como un símbolo: los integrantes tienden a sentirse simbólicamente unidos a la comunidad virtual, creándose una sensación de pertenencia.

2. La Comunidad Virtual como virtual: las comunidades virtuales poseen rasgos comunes a las comunidades físicas, sin embargo el rasgo diferenciador de la comunidad virtual es que ésta se desarrolla, al menos parcialmente, en un lugar virtual, o en un lugar construido a partir de conexiones telemáticas.

La comunidad se distinguirá porque sus integrantes deberán poseer al ingresar en ella, un nivel básico en idioma inglés y en GNU/Linux.

El objetivo común de la comunidad será implementar las ideas de la filosofía de los Unix-BSD, en el desarrollo de aplicaciones teleinformáticas. Esto es avalado por una Certificación BSD que promueve la comunidad internacional. Además intercambiar información (obtener respuestas), conversar y socializar de manera informal a través de comunicación simultánea, ofrecer apoyo y debatir.

La comunicación y un deseo de relación entre los miembros con intereses comunes serán los pilares fundamentales de la comunidad.





Yoandy Pérez Villazón
yvillazon@estudiantes.uci.cu
Proyecto Unicornios



MIGRACIÓN

Herramientas para el acceso a los servicios de Internet

Uno de los principales problemas que afrontan hoy las personas que desean migrar a Software Libre es que existe una gran diversidad de aplicaciones para elegir a la hora de seleccionar qué software vamos a usar cuando se migra de Windows a GNU/Linux, este artículo es resultado de un estudio realizado por el Grupo de Migración, perteneciente al Proyecto de Servicios Especializados para la Migración hacia Software Libre de la Facultad 10 (Unicornios).

Este artículo es tan solo una parte de un conjunto de ellos que hablan sobre las distintas herramientas libres a usar en cada grupo una vez que se decida migrar a Software Libre, en este informe se hablará acerca de las herramientas relacionadas con el menú de internet, es decir aquellas herramientas que comúnmente se utilizan al necesitar algún servicio que esté en la red.

De forma general se abarcan las herramientas relacionadas con :

- Navegadores web.
- Clientes de correo electrónico.
- Lector de noticias RSS.
- Cliente para conectarse a servidores FTP.
- Clientes para los servicios de mensajería instantánea, de manera mas específica sobre el protocolo Jabber.
- Clientes para descargas de archivos bittorrents.
- Clientes de escritorio remoto.

Por último incluye la lista completa de todas las herramientas y una clasificación asignada respecto a sus características.

Las herramientas que se exponen a continuación son las recomendadas para cada recurso que se desee acceder en internet, las mismas han sido probadas y comparadas con otras herramientas sobre Debian GNU/Linux 4.0 y sus características técnicas están bien fundamentadas en el documento [1] emitido por el Grupo de Migración del Proyecto Unicornios de la Facultad 10.

Navegadores de Internet

Navegador	Entorno	Versión
Mozilla Firefox	Cualquiera	2.0
Konqueror	KDE	3.5.8
Elinks	Consola	0.11.1

Principales características de Mozilla Firefox

- Creado por Mozilla Foundation
- Creado en Septiembre del 2002.
- No presenta costo alguno.
- Software Licenciado bajo las licencias MPL, LGPL, GPL.
- Presenta soporte para todas las plataformas, excepto para MAC OS 9
- Presenta administrador de marcadores.
- Posee un gestor de descargas.
- Posee un administrador de contraseñas.
- Posee administrador de formularios.
- Contiene corrección ortográfica.
- Presenta navegación por pestañas.
- Contiene un sistema para el bloqueo anti-popup.
- Presenta zoom de página permitiendo expandir y reducir la página.
- Contiene teclas de acceso HTML.
- Navegación por cursor de texto.
- En cuanto a las tecnologías web Mozilla Firefox soporta : CSS2, Frames, Java, JavaScript, XSLT, XHTML, MathML, Xforms, RSS, Atom.
- En cuanto a los protocolos Mozilla Firefox soporta : FTP, SSL, IRC, Gopher, IDN, data:URL y no soporta el Email.
- En cuanto a los formatos de imagen soporta: JPEG, PNG, GIFs, SVG presentando un soporte parcial para MNG y para PDF.

Principales características de Konqueror

- Creado por KDE.
- Creado en Octubre del año 2000.
- No presenta costo alguno.
- Licenciado bajo GPL.

[1] http://10.33.20.200/Documentacion/Migracion/Herramientas_Migracion_Aplicaciones/



- Soportado para los sistemas GNU/Linux, BSD, UNIX.
- Presenta administrador de marcadores.
- Posee gestor de descargas, utilizando Kget para esta tarea.
- Contiene gestor de descargas para torrents que viene integrado a KDE (Ktorrent).
- Gestiona las contraseñas de acceso con Kwallet.
- Administra los formularios con Kwallet.
- Posee un sistema de corrección ortográfica.
- Presenta navegación por pestañas.
- Contiene un sistema anti-popup.
- Posee un sistema de búsqueda incremental.
- Posee zoom de página que permite expandir y reducir el tamaño de las páginas web.
- Posee teclas de acceso HTML.
- En cuanto al soporte de las tecnologías soporta: CSS2, Frames, Java, JavaScript, XSLT, RSS y ATOM, no soportando MatML, Xforms ni soporte para la autenticación mediante NTLM.
- En cuanto a los protocolos soporta : FTP, SSL, Gopher, IDN, data:url , no presentando soporte para Email, NNTP e IRC.
- En cuanto al soporte para los formatos de imágenes konqueror presenta el soporte para las mas conocidos, como son : JPEG, SVG, GIFs, PNG, PDF.

Principales características de Elinks

- Creado por Baudis Fonseca.
- Creado en diciembre del 2001.
- No presenta costo alguno.
- Licenciado bajo GPL.
- Soportado para todos los sistemas operativos excepto para Mac OS 9.
- Presenta administrador de marcadores.
- Presenta gestor de descargas.
- No posee cliente bittorrent propio.
- Posee administrador de contraseñas.
- No posee corrector ortográfico.
- Posee navegación por pestañas y búsqueda incremental.
- Sobre el soporte de las tecnologías tenemos que soporta frames, no soportando el resto de las tecnologías mencionadas en los navegadores anteriores.
- Sobre el soporte de los protocolos, presenta soporte para: FTP, SSL, NNTP, Gopher, IDN y Data:url, no presentando soporte para Email ni IRC.

- No presenta soporte alguno para formatos de imágenes.

Cientes de correo electrónico

Cliente	Entorno	Versión
Kmail	KDE	4:3.5.5
Evolution	GNOME	2.6.3
Thunderbird	Cualquiera	1.5.0.8

Principales características de Kmail

- Aplicación creada para el entorno de escritorio KDE.
- Licenciado bajo GPL.
- Permite realizar las funciones básicas de un lector de correos.
- Posee carpeta de mensajes.
- Sistema para el trabajo con adjuntos.
- Contiene la posibilidad de incluir contactos y listas de distribución.
- Posee corrector ortográfico.
- Ordenar mensajes bajo distintos criterios.
- Trabajo con libreta de direcciones.
- Permite la utilización de reglas de filtrado.
- Se integra y permite crear con PGP y GnuPG firmas electrónicas y codificar los correos.
- Permite marcar los correos por distintos parámetros.
- Posee un sistema de notificación de correos.

Principales características de Evolution

- Aplicación creada para el entorno de escritorio GNOME.
- Es la herramienta de reemplazo perfecta para Microsoft Exchange.
- Presenta soporte para LDAP, POP, IMAP,SMTP.
- Permite la integración con el calendario iCalendar , el PIM de Mac.
- Presenta soporte para PGP/GPG, SSL y SASL lo que lo hace muy seguro.
- Posee un sistema de detección en tiempo real para el correo basura.
- Permite la creación de carpetas para la recolección de correos bajo distintos criterios.
- Se integra con el cliente de jabber Pidgin (Gaim).
- Puede integrarse a servidores de Microsoft Exchange de correos.

Principales características de Mozilla ThunderBird

- Licenciado bajo MPL, Netscape Public License.
- Permite realizar las funciones básicas de un lector de correos.



- Posee carpeta de mensajes.
- Sistema para el trabajo con datos adjuntos.
- Contiene la posibilidad de incluir contactos y listas de distribución.
- Posee corrector ortográfico.
- Ordenar mensajes bajo distintos criterios.
- Permite marcar los correos por distintos parámetros.
- Posee tratamiento anti-spam para el correo basura, permitiendo gestionar el spam de disímiles maneras.
- Posee una interfaz gráfica por vistas muy configurable que permite colocar el diseño de la interfaz de la manera que deseemos.
- Permite que su correo sea sano y seguro a través de firma digital, cifrado de mensajes, soporte de certificados y dispositivos de seguridad.
- Presenta soporte para IMAP/POP, correo HTML, etiquetas, búsqueda rápida, libreta de direcciones inteligente, filtrado de mensajes avanzado, completado de direcciones LDAP.
- Posee la capacidad de manejar varias cuentas de correo electrónico y noticias.
- Es una aplicación modular que permite extender su funcionamiento mediante módulos.
- Posee sistema de notificación de mensajes.

Lectores de RSS

Lector	Entorno	Versión
Liferea	GNOME	1.0.27
Akreator	KDE	1.2.8
Raggle	Consola	0.0.4

Principales características de Liferea

- Elaborado para el escritorio GNOME.
- Posee carpetas virtuales que permiten clasificar los feeds.
- Posee la capacidad de integrarse a los navegadores.
- Icono en la bandeja del sistema para que cada vez que llegue un nuevo feeds te informe.
- Posee una carpeta de no leídos desde la cual puedes acceder a las noticias no leídos indistintamente de las clasificaciones que estas posean en las distintas carpetas virtuales.

Principales características de Akreator

- Elaborado para el escritorio KDE.
- Posee características similares a Liferea.
- Posee tray-icon que se aloja en la bandeja de entrada del sistema avisándonos cuando llega una nueva noticia.
- Se integra bien con KDE y se puede usar solo o como parte de la suite kcontact.

Principales características de Raggle

- Elaborado con ncurses para interfaz de solo texto.
- Útil para computadores de bajos recursos de hardware.
- Completamente utilizable desde el teclado.
- Se puede usar remotamente a través de una terminal.
- Puede configurarse para que abra los post en un navegador y pestañas por separado.

Cientes para conectarse a servidores FTP

Cliente	Entorno	Versión
FileZilla Client	GNOME	3.0
Kftpgripper	KDE	0.8.0
Midnight Commander	Consola	4.6.1

Principales características de FileZilla

- Presenta criterios de filtrado para la visualización de archivos.
- Licenciado bajo GNU/GPL.
- Presenta un sistema para las actualizaciones automáticas de la aplicación.
- Contiene soporte multi-idioma.
- Soporte para las conexiones sobre firewalls.
- Soporta las conexiones sobre los protocolos FTP, FTPS (TLS/SSL).
- Soporte para recepción y envío de archivos de mas de 4GB.
- Gestión de los limites de la velocidad de transferencia.
- Soporte de arrastrar y soltar para el copiado/movido de elementos.
- Wizard de configuración para los parámetros de la red.

Principales características de Kftpgripper

- Elaborada para el entorno de escritorio KDE.
- Licenciado bajo GNU/GPL.



- Soporte para la apertura de múltiples sesiones FTP (Tabs).
- Contiene cola de transferencia para archivos.
- Soporte para FTP, FTPS (TLS/SSL).
- Soporta el protocolo FXP (Protocolo de intercambio de archivos).
- Soporte para arrastrar y soltar durante el copiado/movido de elementos.
- Posibilidad de contener bookmarks para sitios importantes.
- Integración con kopete para compartir los bookmarks.
- Soporte para importar bookmarks.
- Presenta gráfico estadístico para mostrar el tráfico de transferencias.
- Permite gestionar los límites de las velocidades de transferencias.
- Listas con saltos y de prioridad para las transferencias.
- Permite editar archivos remotos.
- Permite exportar un listado de directorios como archivos de texto.

Principales características de Midnight Commander (Sobre acceso FTP)

- Elaborado en n-curses para ser utilizado en ambientes de consola.
- Licenciado bajo GNU/GPL.
- Permite acceder/editar archivos remotos.
- Posee multi-idioma.
- Permite copiar/mover archivos entre conexiones FTP-FTP, FTP-SSH, FTP-SMB, FTP-Localhost.
- Presenta soporte para FTP, FTPS (TLS/SSL).
- Permite navegación a través de las teclas del cursor.

Cientes de mensajería instantánea sobre el protocolo jabber

Ciente	Entorno	Versión
Gaim	GNOME	2.0.0
Kopete	KDE	3.5.8
Mcabber	Consola	0.8.3.1

Principales características de Gaim

- Elaborada para el entorno de escritorio GNOME.
- Es también conocida como pidgin.
- Licenciado bajo GNU/GPL.
- Es una aplicación multiplataforma.
- Cliente de jabber modular y extensible.
- Soporta múltiples protocolos, entre ellos AIM, MSN, Yahoo!, Jabber, ICQ, IRC, SILC, Novell Groupwise, Lotus Sametime, Zephyr, Gadu – Gadu, QQ.

- Permite agrupar los contactos por grupos.
- Permite visualizar el estado de los contactos.
- Presenta tabs para entablar las conversaciones con los usuarios.
- Permite notificar las acciones de los contactos.
- Permite oír lo que un usuario nos escribe a través de la integración de este con el software festival.
- Diversos métodos de autenticación con los protocolos soportados permite la autenticación segura y a través de servidores proxy.
- Permite el envío y recepción de archivos.
- Permite la inserción de íconos de emociones.
- Permite la corrección ortográfica.

Principales características de Kopete

- Elaborada para el entorno de escritorio KDE.
- Licenciado bajo GNU/GPL.
- Cliente de jabber modular y extensible.
- Soporta múltiples protocolos, entre ellos .Net Messenger Service, AOL Instant Messenger, Gadu – Gadu, ICQ, Internet Relay Chat, Jabber, Lotus Sametime, Novell Groupwise, SMS, Skype, WinPopup, Yahoo! Messenger.
- Permite agrupar los contactos por grupos.
- Permite visualizar el estado de los contactos.
- Presenta tabs para entablar las conversaciones con los usuarios.
- Permite notificar las acciones de los contactos.
- Se integra con las demás aplicaciones del escritorio KDE como KaddressBook y Kmail.
- Diversos métodos de autenticación con los protocolos soportados permite la autenticación segura y a través de servidores proxy.
- Permite el envío y recepción de archivos.
- Permite la inserción de íconos de emociones.
- Permite la corrección ortográfica.
- Permite insertar estilos en las conversaciones utilizando XSL y CSS.

Principales características de mcabber

- Elaborado en n-curses para el entorno de consola.
- Licenciado bajo GNU/GPL.
- Cliente ligero de jabber.
- Presenta soporte para SSL.



- Permite chatear con múltiples usuarios.
- Almacena en el historial las conversaciones de los usuarios.
- Permite el auto-completamiento con < tab > al igual que GNU/Linux.
- Permite el aviso de nuevos mensajes de usuarios, siendo esta última opción configurable desde scripts externos.

Cientes para la descarga archivos BitTorrents

Ciente	Entorno	Versión
Gnome-btdownload	GNOME	0.0.25
Ktorrent	KDE	2.0.3
rtorrent	Consola	0.6.4.1

Principales características de Gnome-btdownload

- Elaborada para el entorno de escritorio GNOME.
- Licenciado bajo GPL.
- Altamente integrado al entorno GNOME.
- Capaz de ejecutar hasta 8 torrents activos.
- Libre de malware y software dañinos.
- Programado en Python.
- Permite realizar el proceso de descarga de torrents de manera muy sencilla.

Principales características de ktorrent

- Elaborada para el entorno de escritorio KDE.
- Licenciado bajo GPL.
- Herramienta libre de software malicioso.
- Admite UPnP Port Mapping.
- Admite protocolo de cabecera encriptado.
- Desarrollado en C++.
- Admite descarga selectiva.
- Está integrada en el navegador web.

Principales características de rtorrent

- Elaborada para el entorno de consola.
- Licenciado bajo GPL.
- Es portable a Mac OS y GNU/Linux.
- Herramienta libre de software malicioso.
- Admite priorización.
- Admite descargas selectivas.

Cientes para la conexión a través del escritorio remoto

Ciente	Entorno	Versión
tsclient	GNOME	0.148
krdc	KDE	3.5.5

Principales características de Linux Terminal Server Client (tsclient)

- Elaborada para el entorno de escritorio GNOME.
- Licenciado bajo GPL.
- Usa el programa rdesktop como base.
- Permite el acceso a Windows NT/2000/XP.
- Admite la lectura de archivos .rdp escritos en el formato unicode de Microsoft.
- Permite conectarse a servidores VNC, ya que contiene el cliente vncviewer.
- Permite conectarse al cliente ICA de CITRIX.
- Permite al acceso a máquinas que sirvan XDMCP en la red a través de Xnest.

Principales características de krdc

- Elaborada para el entorno de escritorio KDE.
- Licenciado bajo GPL.
- Permite el acceso a servidores con protocolo rfb a través de VNC.
- Utiliza rdesktop como base para el acceso a Windows NT/2000/XP, utilizando RDP.
- No es tan configurable como Linux Terminal Server Client.

Con este artículo como material de estudio el lector será capaz de:

- Elegir la mejor herramienta para acceder o manejar un determinado recurso en internet,
- Poder visualizar las principales características de las herramientas más importantes en cada tema
- Acceder a un listado completo de todas la herramientas libres que existen para realizar la tareas de navegación y uso de los recursos de internet.



David Padrón Álvarez
dpadron@estudiantes.uci.cu
PROYECTO GRAPh TOol



SOLUCIÓN

GRUB en un disquete

El Dual Boot o Arranque Dual son distintas formas de llamar a la capacidad de una computadora de poder tener más de un Sistema Operativo (SO) y poder iniciar con cualquiera de ellos.

Al arrancar la computadora con doble booteo, una pantalla preguntará al usuario cuál de los sistemas instalados quiere utilizar, y pasada esta etapa comenzará la carga de sólo el sistema elegido. La capacidad de seleccionar el sistema a arrancar está otorgada por el *Cargador o Gestor de arranque (Boot Loader)*.

El GRUB (**GR**and **U**nified **B**ootloader) es un Gestor de arranque desarrollado por el Proyecto GNU y que hoy día viene preinstalado en la mayoría de las distribuciones de GNU/Linux modernas.

El Dual Boot se ha convertido en algo esencial, sobre todo para aquellos usuarios que comienzan a usar GNU/Linux o comparten la PC con otras personas que necesitan inevitablemente el uso de programas hechos para la plataforma del SO Windows.

Este artículo no pretende ser una presentación del GRUB y sus características, sino una guía para solucionar un problema que puede surgir cuando se comparten Windows y GNU/Linux en una sola computadora.

Para los que se deciden por esta variante de tener los dos SO por primera vez, lo ideal es instalar Windows y luego GNU/Linux; pues este último es capaz de reconocer la existencia de otro SO previamente instalado y autoconfigurar su archivo **menu.lst** que se encuentra en la ruta **/boot/grub** y que contiene las especificaciones para esta pantalla del Gestor de arranque; algo que no es capaz de hacer el Sistema de Bill Gates en su **boot.ini**.

Hasta aquí todo va bien, pero el problema surge cuando necesitamos instalar Windows en una PC que sólo tiene GNU/Linux sin borrarlo, o cuando necesitamos actualizar Windows a una nueva versión en la PC con dual boot.

En este caso, el gigante de la Microsoft no implementó nada que permitiera una actualización casi transparente como en GNU/Linux. Por lo tanto, cualquier modificación de Windows traería consigo la reescritura del **Master Boot Record (MBR)** y la inutilización del GRUB.

Cuando el cargador de arranque de Windows sustituye al de GNU/Linux y se desea arreglarlo, hay una herramienta que lo permite: **Super Grub Disk**. Está disponible en Internet, pero para los que tienen limitaciones con la navegación o simplemente prefieren *tirar* líneas en el terminal, he seleccionado estos 10 comandos para tener el GRUB en un disquete, algo verdaderamente muy útil.

Con un disco floppy de 1.44Mb puesto en la unidad se teclea lo siguiente:

```
fdformat /dev/fd0

sudo mkfs /dev/fd0

sudo mount /media/floppy0

sudo mkdir -p /media/floppy/boot

sudo cp -r /boot/grub/media/floppy/boot

sudo umount /media/floppy

sudo grub

root (fd0)

setup (fd0)

quit
```

Luego de haber entrado nuevamente a la distro de GNU/Linux con el disquete, se usa el comando:

```
sudo grub-install /dev/hd0
```



para volver a instalar el GRUB en el disco duro. O de la siguiente manera con el intérprete de comandos del GRUB:

```
sudo grub
root (hdX,Y)
setup (hdX)
quit
```



(Donde X es el número del disco duro y Y la partición donde está instalado en GNU/Linux, siempre tomando que estas se empiezan a numerar por 0)

Eiger Mora Moredo
emora@estudiantes.uci.cu
Proyecto Unicornios



Los que somos viejos en el mundo del pingüino, siempre solíamos arreglar este problema con cualquier LiveCD de Knoppix, pero esta solución es sin dudas más rápida y eficiente, pues permite editar el archivo **menu.lst** de este GRUB portable, en cualquier momento.

Para más información sobre el GRUB y su configuración, pueden visitar la página: <http://www.guia-ubuntu.org/index.php?title=GRUB>

“Cómo construir un GrubSplash”

En la prisa y el afán de configurar GNU/Linux al gusto y necesidades de cada cual, se hecha a rodar la imaginación buscando la imagen perfecta de la PC. El Grub es, la cara de nuestro sistema, y de nada nos vale configurar el bootsplash y la interfaz de los gestores y entornos de escritorio si, el portal de la casa, está tétrico. Este artículo tiene como objetivo enseñar a construir un GrubSplash utilizando GIMP para “ponerle” al Grub la cara que se desee.

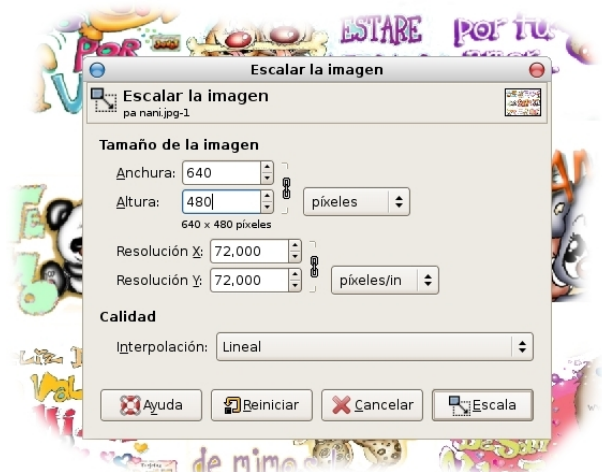
Escogiendo la Imagen

Se puede diseñar una imagen, escogerla del “almacén de fotos” o simplemente bajarla de Internet, eso si, para cuidar desde ahora la calidad de esta no se debe utilizar ninguna que contenga muchos gradaciones de colores.

Ajustando las propiedades de la Imagen

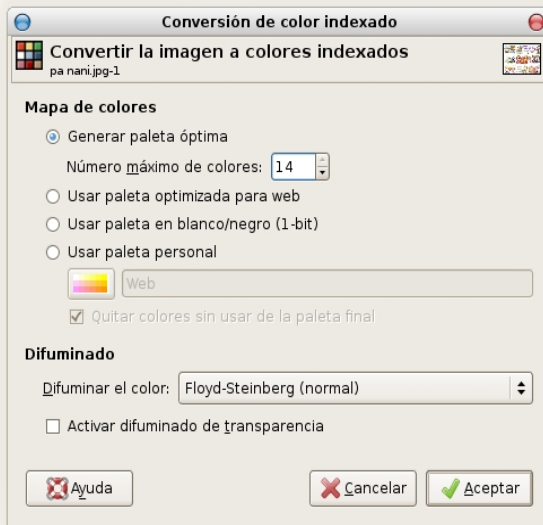
La imagen debe cumplir con varias propiedades:

1. Resolución: 640x480





2. Número de colores: 14



3. Formato: xpm



Configurando el Grub

Lo ideal es guardar las imágenes en alguna carpeta dentro del directorio `/boot/grub/` con un nombre que haga referencia (Ej. `images`)

Una vez listas las imágenes en formato `xpm` se comprimen con `gzip`:

1. En la consola: `# gzip image.xpm > image.xpm.gz`

2. Interfaz gráfica: Se utiliza el gestor de archivos comprimidos (`file-roller`, `ark`, `xarchiver`).

Ya comprimidas se copian para `/boot/grub/images`(si se siguió el ejemplo) y se configura el Grub, para esto se modifica el archivo `menu.lst` adicionándole la siguiente línea:

```
splashimage=(hd0,#)/boot/grub/images/image.xpm.gz
```

```
1. title Debian GNU/Linux, kernel
2.6.18-5-486
2. root (hd0,4)
3. kernel /boot/vmlinuz-2.6.18-5-486
root=/dev/sda5 ro vga=791
4. initrd /boot/initrd.img-
2.6.18-5-486
5. savedefault
```

#: Es el número de la línea root, para el caso del ejemplo (hd0, 4)

Importante: Nótese que la línea que se adiciona está resaltada en otro color (verde).

Listo, se guardan las modificaciones al `menu.lst`, se reinicia y se podrá disfrutar del nuevo "look" del Grub.





Dayron Pérez Roldán
droidan@estudiantes.uci.cu
Proyecto Unicornios



Programación de hilos y semáforos en GNU/Linux con C++

Un **hilo de ejecución**, en sistemas operativos, es una característica que permite a una aplicación realizar varias tareas concurrentemente. Un **semáforo** es una variable especial protegida (o tipo abstracto de datos) que constituye el método clásico para restringir o permitir el acceso a recursos compartidos. En este artículo se abordan los aspectos más importantes de la programación de hilos y semáforos, utilizando Code::Blocks como entorno de Programación para escribir el código, el cual podría ser escrito en cualquier IDE (Entorno de Desarrollo Integrado) de programación, que soporte C++.

En la medida que se desarrollan aplicaciones pueden encontrarse ciertos requisitos que los programas deben cumplir para ser más eficaces y óptimos, normalmente las aplicaciones se harán más complejas pero se encontrarán vías óptimas para dar solución a diferentes tipos de problemas. Estos problemas se pueden plantear más específicamente en el plano de la programación paralela que es una técnica que enfatiza la ejecución simultánea [1] de diferentes tareas ya sea en un mismo ordenador o en un cluster de ordenadores.

El mayor problema de la programación paralela radica en la complejidad de sincronizar unas tareas con otras, ya sea mediante secciones críticas, semáforos o paso de mensajes, para garantizar la exclusión mutua en las zonas del código en las que sea necesario.

Es importante aclarar que un hilo de ejecución no es lo mismo que un proceso; la diferencia que existe entre ambos es que un proceso en Unix es cualquier programa en ejecución y es totalmente independiente de otros procesos el cual tiene su propia zona de memoria y se ejecuta "simultáneamente" a otros procesos. En Unix resulta casi imposible que un proceso se introduzca en la zona de memoria de otro proceso, además de que si el proceso se cae, se cae solo él.

Ahora bien, dentro de un proceso pueden haber varios hilos de ejecución (varios threads). Eso quiere decir que un proceso podría estar haciendo varias cosas "a la vez".

Los hilos dentro de un proceso comparten todos la misma memoria. Quiere decir que si un hilo toca una variable, todos los demás hilos del mismo proceso verán el nuevo valor de la variable.

Esto hace imprescindible el uso de semáforos o mutex (Exclusión Mutua, que en inglés es al revés, funciones pthread_mutex) para evitar que dos threads accedan a la vez a la misma estructura de datos.

También hace que si un hilo "se equivoca" y corrompe una zona de memoria, todos los demás hilos del mismo proceso vean la memoria corrompida. Un fallo en un hilo puede hacer fallar a todos los demás hilos del mismo proceso.

Un proceso es, por tanto, más costoso de lanzar, pues se necesita crear una copia de toda la memoria del programa. Los hilos son más ligeros.

En cuanto a complejidad, en los hilos, al compartir la memoria y los recursos, es casi obligado el uso de mutex o semáforos, así que su programación suele ser más complicada y se necesita ser más cuidadoso.

¿Qué elegir: un proceso o un hilo? Depende de muchos factores, el autor suele elegir procesos cuando una vez lanzado el hijo no requiere demasiada comunicación con él. Elige hilos cuando tienen que compartir y actualizarse datos.

En caso de necesidad de gestionar entradas/salidas es mejor procesos (atender simultáneamente a varias entradas de Sockets, por ejemplo), y para hacer programas con muchos cálculos en paralelo con varias CPU es mejor hilos, siempre y cuando el sistema operativo sea capaz de repartir automáticamente los hilos en las distintas CPU en función de su carga de trabajo.

Para desarrollar este artículo como ya bien se ha dicho se abundará un poco más en la programación de hilos y por supuesto acompañado de los semáforos.

[1] El sistema operativo irá ejecutando pedazos de programas por turnos de forma muy rápida, dando la sensación de simultaneidad.



Para hacer un poco más sencillo de comprender este artículo se dividirá la siguiente explicación en dos partes: Hilos y Semáforos. Y finalmente se ilustrará a través de un ejemplo lo antes explicado.

Hilos:

Lo que es propio de cada hilo es el contador de programa, la pila de ejecución y el estado de la CPU (incluyendo el valor de los registros).

Algunos lenguajes de programación tienen características de diseño expresamente creadas para permitir a los programadores lidiar con hilos de ejecución (como Java). Otros (la mayoría) desconocen la existencia de hilos de ejecución y éstos deben ser creados mediante llamadas de biblioteca especiales que dependen del sistema operativo en el que estos lenguajes están siendo utilizados (como es el caso del C y del C++).

Un ejemplo de la utilización de hilos es tener un hilo atento a la interfaz gráfica (iconos, botones, ventanas), mientras otro hilo hace una larga operación internamente. De esta manera el programa responde de manera más ágil a la interacción con el usuario. También pueden ser utilizados por una aplicación servidora para dar servicio a múltiples clientes.

Otro ejemplo puede ser en un programa de hoja de cálculo un hilo puede estar visualizando los menús y leer la entrada del usuario mientras que otro hilo ejecuta los órdenes y actualiza la hoja de cálculo.

Los elementos asíncronos de un programa se pueden implementar como hilos. Un ejemplo es como los software de procesamiento de texto guardan archivos temporales cuando se está trabajando en dicho programa. Se crea un hilo que tiene como función guardar una copia de respaldo mientras se continúa con la operación de escritura por el usuario sin interferir en la misma.

Se pueden implementar dos tipos de hilos:
Hilos a nivel de usuario.
Hilos a nivel de Kernel.

También conocidos como ULT (User Level Thread) y KLT (Kernel Level Thread).

Estados de un hilo:

Los principales estados de los hilos son: Ejecución, Listo y Bloqueado. No tiene sentido asociar estados de suspensión de hilos ya que es un concepto de proceso.

En todo caso, si un proceso está expulsado de la memoria principal (ram), todos sus hilos deberán estarlo ya que todos comparten el espacio de direcciones del proceso.

Cambio de estados

Creación: Cuando se crea un proceso se crea un hilo para ese proceso. Luego, este hilo puede crear otros hilos dentro del mismo proceso. El hilo tendrá su propio contexto y su propio espacio de pila, y pasará a la cola de listas.

Bloqueo: Cuando un hilo necesita esperar por un suceso, se bloquea (salvando sus registros). Ahora el procesador podrá pasar a ejecutar otro hilo que esté en la cola de Listos mientras el anterior permanece bloqueado.

Desbloqueo: Cuando el suceso por el que el hilo se bloqueó se produce, el mismo pasa a la cola de Listos.

Terminación: Cuando un hilo finaliza se liberan tanto su contexto como sus pilas.

Los sistemas operativos generalmente implementan hilos de dos maneras:

Multihilo apropiativo: permite al sistema operativo determinar cuándo debe haber un cambio de contexto. La desventaja de esto es que el sistema puede hacer un cambio de contexto en un momento inadecuado, causando un fenómeno conocido como inversión de prioridades y otros problemas.

Multihilo cooperativo: depende del mismo hilo abandonar el control cuando llega a un punto de detención, lo cual puede traer problemas cuando el hilo espera la disponibilidad de un recurso.

El soporte de hardware para multihilo desde hace poco se encuentra disponible. Esta característica fue introducida por Intel en el Pentium 4, bajo el nombre de HyperThreading.

En GNU/Linux las funciones que permiten crear un nuevo hilo de ejecución es **pthread_create** () que admite cuatro parámetros:

pthread_t * es un puntero a un identificador de thread.

pthread_attr_t * son los atributos de creación del hilo.

void (*) (void *) es la función que se ejecutará como un hilo aparte.

void * es el parámetro que se le pasará a la función anterior cuando se ejecute en el hilo aparte.



A veces es necesario hacer que un hilo espere a otro hilo. Por ejemplo, supongamos que varios hilos están realizando un cálculo y es necesario el resultado de todos ellos para obtener el resultado total. El hilo encargado de este resultado total debe esperar a que todos los demás hilos terminen. Sin embargo, no todos los hilos son susceptibles de ser esperados. Hay hilos a los que el sistema libera automáticamente todos sus recursos cuando terminan. Con otros, sin embargo, no lo hace.

Que el sistema haga una cosa u otra, depende de los atributos que se le hayan pasado al crear el hilo con `pthread_create()`. Por los primeros hilos, los que se liberan automáticamente, no se puede esperar. Por los segundos, los que mantienen sus recursos, sí se puede esperar. De hecho, si algún hilo no espera a que termine, sus recursos nunca se liberarán (salvo que se liberen explícitamente con la función `pthread_detach()`).

Si se ponen los atributos de creación a `NULL`, el hilo es por defecto "esperable". De todas formas, se pondrá explícitamente el atributo para ver cómo funciona.

Los atributos de un hilo son de tipo `pthread_attr_t`. Para rellenar esta estructura con sus valores por defecto, se tiene la función `pthread_attr_init()`. El código para obtener los atributos con su valor por defecto es:

```
pthread_attr_t    atributos;
...
pthread_attr_init (&atributos);
```

Ahora se puede cambiar cualquiera de los campos de esta estructura con funciones específicas para ello. En concreto, interesa la función `pthread_attr_setdetachstate()`, que permite cambiar esta característica. Hay dos valores posibles para el atributo, `pthread_create_joinable` y `pthread_create_detached`. Con el primer valor puede esperarse por el hilo, con el segundo se libera automáticamente. El código quedaría:

```
pthread_attr_setdetachstate
(&atributos, pthread_create_joinable);
```

Con esto es posible crear un thread al que se puede esperar con la función `pthread_create()`. Este thread termina cuando sale de la función que está ejecutando o llamando a `pthread_exit()`.

```
pthread_exit ((void *)"Fin");
```

El parámetro que se pasa es el valor que se devolverá al hilo que espera. Como es un `void *`, puede ser un puntero a lo que se quiera.

El hilo que espera, debe llamar a la función `pthread_join()`. Esta llamada hace que el hilo se "duerma" hasta que el otro hilo termine. Si el otro hilo ya había terminado, la función `pthread_join()` sale inmediatamente. La función `pthread_join()` admite dos parámetros. El primero es el identificador `pthread_t` del hilo por el que se quiere esperar. El segundo es un `void **`. Si le pone algo que no es `NULL`, en él devolverá lo que ha devuelto el hilo hijo. En este ejemplo, como se sabe que devolverá un `char *`, se declarará y eso será lo que se pase.

```
pthread_t idHilo;
char *valorDevuelto = NULL;
...
pthread_join (idHilo, (void**)
&valorDevuelto);
```

Semáforos:

A veces es necesario que dos o más procesos o hilos (threads) accedan a un recurso común (escribir en un mismo fichero, leer la misma zona de memoria, escribir en la misma pantalla, etc). El problema es que si lo hacen simultáneamente y de forma incontrolada, pueden "machacar" el uno la operación del otro (y dejar el fichero o la memoria con un contenido inservible o la pantalla ilegible).

Para evitar este problema, están los semáforos. Un semáforo da acceso al recurso a uno de los procesos y se lo niega a los demás mientras el primero no termine. Los semáforos se emplean para permitir el acceso a diferentes partes de programas (llamados secciones críticas) donde se manipulan variables o recursos que deben ser accedidos de forma especial. Según el valor con que son inicializados se permiten a más o menos procesos utilizar el recurso de forma simultánea.

Existen actualmente dos problemas muy conocidos donde sus soluciones se basan en el uso de semáforos, estos son:

[El problema del barbero durmiente.](#)

[El problema de los filósofos cenando.](#)



El funcionamiento del semáforo es como el de una variable contador. Imaginemos que el semáforo controla un fichero y que inicialmente tiene el valor 1 (está "verde"). Cuando un proceso quiere acceder al fichero, primero debe decrementar el semáforo. El contador queda a 0 y como no es negativo, deja que el proceso siga su ejecución y, por tanto, acceda al fichero.

Ahora un segundo proceso lo intenta y para ello también decrementa el contador. Esta vez el contador se pone a -1 y como es negativo, el semáforo se encarga de que el proceso quede "bloqueado" y "dormido" en una cola de espera. Este segundo proceso no continuará por tanto su ejecución y no accederá al fichero.

Suponga ahora que el primer proceso termina de escribir el fichero. Al acabar con el fichero debe incrementar el contador del semáforo. Al hacerlo, este contador se pone a 0. Como no es negativo, el semáforo se encarga de mirar en la cola de procesos pendientes y "desbloquear" al primer proceso de dicha cola. Con ello, el segundo proceso que quería acceder al fichero continúa su ejecución y accede al fichero. Cuando este proceso también termine con el fichero, incrementa el contador y el semáforo vuelve a ponerse a 1, a estar "verde".

Acceso sincronizado a recursos comunes con en el trabajo con hilos.

Para sincronizar están los mutex. Un mutex es como un semáforo que deja o no pasar al hilo. Antes de acceder a la estructura de datos, el hilo debe mirar el semáforo. Si el semáforo está rojo, el hilo se queda bloqueado hasta que se ponga verde. Si está verde, el hilo podrá acceder a la estructura de datos y el semáforo se pone rojo en cuanto pase este hilo. Una vez que termine de acceder a la estructura de datos, el hilo debe acordarse de poner verde el semáforo. Para crear uno de estos semáforos existe la función `pthread_mutex_init()`. A esta función se le pasa un `pthread_mutex_t *` que luego servirá de identificador del semáforo (se pueden crear tantos como se necesiten). Un segundo parámetro de la función son los atributos de creación del semáforo. Vale poner **NULL** para un comportamiento por defecto. El código para crear un mutex es:

```
pthread_mutex_t mutexBuffer;  
...  
pthread_mutex_init (&mutexBuffer,  
NULL);
```

Con el comportamiento por defecto, un mutex puede ponerlo en verde cualquier hilo, no necesariamente el que lo ha puesto rojo. Se puede cambiar este comportamiento con el segundo parámetro de `pthread_mutex_init ()`. Con el comportamiento por defecto, sólo tiene efecto el poner rojo o verde el semáforo. Se puede cambiar este comportamiento para que se pueda poner rojo varias veces seguidas, de forma que cada vez se pone "más rojo". Para que finalmente se ponga verde, hace falta ponerlo verde tantas veces como se puso rojo anteriormente. Esto es un comportamiento recursivo y sirve para hilos que utilicen recursividad.

`pthread_mutex_init ()` siempre devuelve 0.

Una vez inicializado el mutex, cuando un hilo quiera acceder a una estructura de datos que esté protegida por este mutex, debe llamar a la función `pthread_mutex_lock ()`, pasándole el identificador del mutex. Si el semáforo está verde, se pone rojo y la función retorna inmediatamente, con lo que el hilo, en las siguientes líneas de código puede acceder a la estructura de datos. Cuando termine con la estructura de datos, debe volver a poner verde el semáforo, llamando a la función `pthread_mutex_unlock ()`, pasando de nuevo el identificador del mutex. El código quedaría:

```
pthread_mutex_lock (mutexBuffer);  
/*aquí se accede a la estructura de  
datos */  
pthread_mutex_unlock (mutexBuffer);
```

La función `pthread_mutex_lock ()` puede bloquear al hilo hasta que alguien ponga el semáforo verde y esto, puede ser para toda la vida (depende de los otros hilos). Si es muy importante que un hilo no se quede bloqueado esperando un mutex, se puede llamar a la función `pthread_mutex_trylock ()`, que se comporta igual que `pthread_mutex_lock ()`, pero no deja bloqueado al hilo en caso de estar el semáforo rojo, sino que devuelve un error. El hilo que llama a esta función, debe verificar si se ha producido ese error o no antes de acceder a los datos.

Con esto queda todo listo. Cuando no se necesite más el mutex, se libera con `pthread_mutex_destroy ()`.

Explicados todos estos detalles de la programación de hilos y semáforos se está en condiciones de pasar a ver un ejemplo.



Lo siguiente es un ejemplo de cómo dos hilos esperan el uno por el otro para acceder a los datos de un array de enteros. Uno de los hilos (el del main) pone todos los items del array con el valor 0, luego los rellena todos con el valor 1, después 2 y así sucesivamente. El otro hilo verifica que todos los valores sean iguales al de la posición 0, es decir, todos iguales, dando un error si no lo son.

```
#include <pthread.h>
#include <iostream>

Void *funcionThread (void *parametro);

#define tamano_buffer 1000
Into buffer [tamano_buffer];
Pthread_mutex_t mutexBuffer;

int main()
{
    Pthread_t idHilo;
    Int error;
    Int contador = 0;
    Int i;
    pthread_mutex_init (&mutexBuffer,
NULL);

    Error = pthread_create (&idHilo,
NULL, funcionThread, NULL);

    If (error != 0)
    {
        perror ("No puedo crear
thread");
        Exit (-1);
    }

    While (1)
    {
        pthread_mutex_lock
(&mutexBuffer);

        for (i=0; i<tamano_buffer;
i++)
            buffer[i] = contador;

        pthread_mutex_unlock
(&mutexBuffer);
        contador++;
    }
    return 0;
}

void *funcionThread (void *parametro)
{
    int i;
    int elementoDistinto = 0;

    while (1)
    {
        pthread_mutex_lock
(&mutexBuffer);
```

```
for (i=1; i<tamano_buffer
i++)
    {
        if (buffer[0] !=
buffer[i])
            {
                ElementoDistinto = 1;
                break;
            }
        If (elementoDistinto)
            printf ("Hijo :
Error. Elementos de buffer distintos\n");
        else
            printf ("Hijo :
Correcto\n");

            ElementoDistinto = 0;

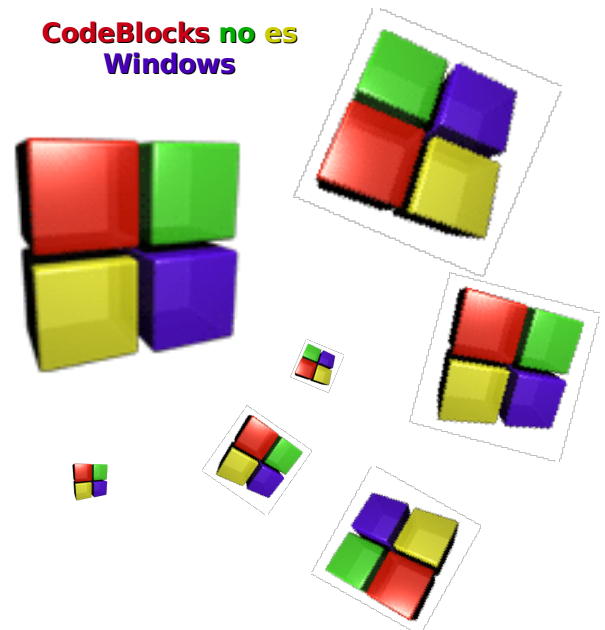
            pthread_mutex_unlock
(&mutexBuffer);
    }
}
```

En este ejemplo se utilizaron los semáforos y todo debe estar correcto, pero si no se hubiesen usado y se quitaran del código, al estar los dos hilos ejecutándose de forma independiente, es posible que uno intente acceder (leer o escribir) el arreglo de datos cuando el otro hilo no ha acabado (de leer o escribir) en dicha arreglo.

Si, por ejemplo, uno de ellos empieza a escribir datos en el arreglo y cuando está a medias el otro hilo se pone a leer, este segundo leerá datos incoherentes.

Con este artículo se ha intentado abordar los temas más importantes de la teoría de hilos y semáforos y se ha expuesto un ejemplo para ilustrarlo. No se pretende con esto enseñar a programar hilos y semáforos pero si lograr una cultura general del tema.

CodeBlocks no es Windows





ENTREVISTA ¿X?

Entrevista a:
Marcos Luis Ortiz Valmaseda
Por: **Ismarai Nuñez Viltres**
iviltres@estudiantes.uci.cu
Movimiento "Código y Letra"



Comunidad de Software Libre por la universalización del conocimiento

Fue en ardua labor organizativa, debido a la celebración del primer aniversario de la Revista UXi, en que tuve la grata ocasión de conocer a los integrantes de la Comunidad de Software Libre. Disciplina, responsabilidad, unidad, son solo algunas de las impresiones que me causó aquel breve encuentro.

Al calor de sus 5 años de fundada, esta comunidad ha reunido usuarios interesados en los temas de GNU/Linux en un contorno donde convergen todas las ideas en una misma dirección: impulsar el uso del software libre, primeramente en toda la universidad y luego en todo el país.

La entrevista a Marcos Ortiz Valmaseda, coordinador de la comunidad de Software Libre de la UCI, me dio la oportunidad de conocer aun más sobre sus temas polémicos y los hitos alcanzados.

-Luego del su surgimiento en el 2003, ¿Cuál fue el primer logro de la comunidad?

-Un punto a destacar fue el lanzamiento en el año 2004 del Portal de Software Libre versión 1.0, el cual constituyó el pilar clave de la Comunidad en la Universidad, por el que llegaban todas las informaciones importantes referentes al tema, además del aprovechamiento de los foros de discusión para darle respuesta a las interrogantes de los más novatos en el uso de GNU/Linux.

-Coméntame sobre las misiones, visiones y proyectos futuros.

-La misión fundamental de la Comunidad de Software Libre es impulsar el uso del mismo en toda la universidad. Actualmente forman parte de dicha comunidad alrededor de 1490 usuarios, entre estudiantes, profesores y trabajadores.

La meta fundamental que perseguimos es seguir creciendo como Comunidad, y así integrarnos todos al desarrollo de nuestra distribución: NOVA GNU/Linux, para crear una plataforma base para la migración de la universidad y el país en conjunto, lo que

permite alcanzar la independencia tecnológica que tanto perseguimos y le inculcamos a los usuarios que vienen de Windows.

Tenemos muchos planes futuros, pero lo primero que viene ahora, es la preparación de la migración del MINED, organismo que nos dará mucho prestigio, no sólo a nivel nacional, sino internacional también, ya que la tarea será verdaderamente ardua.

-¿Cuáles según tu consideración, son los mayores retos que se les avecinan?

-La prueba de fuego de nuestra distribución NOVA GNU/Linux, con la cual se llevará a cabo la migración de ETECSA, permitirá demostrar que la misma puede ser adaptada a cualquier entorno, y la migración de la Fiscalía General de la República que también está abogando por el SWL, por lo que cada día tenemos que abogar más por la migración de la universidad.

-¿Mantienen vínculo con otras comunidades universitarias y el país?

-La Comunidad, sobre todo el equipo de desarrollo de NOVA, mantiene estrechas relaciones con otras comunidades de desarrollo que trabajan con otras distribuciones basadas en Gentoo (metadistribución en la cual está basada NOVA) también como Sabayón, Ututo, además del mismo Daniel Robbins, creador de Gentoo, el cual está haciendo un gran aporte en la parte de Server de NOVA.

Con la comunidad cubana de software libre se tienen muy pocos lazos, ya que no se unen en un mismo concepto, siempre terminan con discusiones vanas de que "cuál distribución yo uso" y no abogan por la visión de que la diversidad en el Software Libre es una ventaja sobre los sistemas operativos propietarios y que no están atados a una sola pauta o criterio.



-Es apreciable el crecimiento de usuarios que usan el Software Libre en la universidad. Las facultades regionales de Granma y Ciego de Ávila usan sólo Software Libre; y en la de Artemisa, ya están migrados los laboratorios de producción, trabajando a un 100%. ¿Qué otros resultados relevantes se pueden destacar?

-Otros resultados son la migración exitosa de las Oficinas Adjuntas del Consejo de Estado "Ira y B", los cuales están muy satisfechos con este hecho; la consultoría de muchos organismos dentro y fuera del país que han abogado por el Software Libre ganando una gran cantidad de prestigio en el mundo; la culminación exitosa de las dos primeras fase del Proyecto RINDE con Venezuela; el surgimiento de la Revista de Software Libre de la UCI: UXi, la cual en tan sólo un año de vida ha ganado no sólo adeptos en el terreno nacional, sino también en el internacional.

Una gran cantidad de servicios a disposición de la Comunidad como: repositorios de variadas distribuciones (NOVA, Gentoo, Debian, Ubuntu, Arch Linux, Red Hat, Fedora y Open SUSE), la versión 2.0 del Portal de Software Libre, muy esperado por la comunidad, etc. Además de la personalización de NOVA para usarlo como plataforma de desarrollo de aplicaciones informáticas (BioNova) con una gran aceptación del proyecto.

-¿Qué impacto tuvo FLISOL?

-En la universidad no tuvo el impacto que tuvo en América Latina, pero sí se acometieron varias acciones como conferencias, charlas, festivales de instalación de distribuciones, etc. Esperemos que para el próximo haya un poco más de auge por parte de la comunidad.

-¿Como ha sido el apoyo recibido de la dirección universitaria a los objetivos y proyectos de la comunidad?

-Es importante destacar que el Software Libre en cualquier lugar enfrenta resistencia al cambio en un proceso de migración, nuestra comunidad no está ajena a ello. En la Universidad aún no existe una conciencia del cambio de concepto que supone la migración.

La dirección de la UCI nos ha apoyado y entienden la importancia de una migración, aunque nosotros consideramos que este apoyo no ha sido el suficiente y muchas veces nos enfrentamos a trabas y obstáculos. No obstante hemos demostrado que usar software libre no es un capricho de los comunitarios, sino una necesidad de alcanzar la soberanía tecnológica.

Actualmente tenemos mucho más apoyo que antes, pero todavía es insuficiente.

HUMOR LIBRE



TIRA ECOL (CC some rights reserved) - Javier Malonda



[Version Español] <http://tira.escomposlinux.org>



[English Version] <http://comic.escomposlinux.org>



TIRA ECOL (CC some rights reserved) - Javier Malonda



[Versión Original] <http://www.tiraecol.net>



[English Version] <http://en.tiraecol.net>



Marisniulkis Lescaille Cos
mlescaille@estudiantes.uci.cu
Facultad 2



Apache como servidor Web por excelencia

En la publicación número 9 de la revista en esta misma sección y con el tema "Servicios telemáticos sobre GNU/Linux" se anunciaba una serie de artículos relacionados con este tópico. Sin intención de establecer un orden jerárquico comenzaremos por el servidor Web Apache.

Alojar hoy un sitio Web (Web hosting) no es problema. Por toda la red se pueden encontrar anuncios de alojamiento Web sobre diferentes plataformas, especialmente sobre Linux alegando la seguridad del sistema operativo. Un servidor Web es un programa que implementa el protocolo HTTP (HyperText Transfer Protocol). Este protocolo está diseñado para transferir las páginas Web. Los servidores Web almacenan sitios Web así como otros archivos asociados a estos como imágenes, páginas Web, textos, sonidos, bases de datos, etc.

Permiten brindar la información contenida en estos sitios de forma pública o restringida a los usuarios. Al referirse al término servidor, generalmente, no sólo se asocia al programa sino también a la máquina donde se encuentra. Estas computadoras deben poseer altos requerimientos de velocidad, memoria y espacio en disco para que sean capaces de no colapsar por recursos ante un número grande de peticiones. No obstante, Apache no requiere grandes recursos para funcionar, en sistemas GNU/Linux correrá bien con 5 a 12 MB de disco duro y 8 MB de memoria RAM.

¿Qué es Apache?

Apache se inició como iniciativa basada en el servidor NCSA (National Center for Super Computer Applications) http 1.3 a principios de 1995, luego fue reescrito todo su código y actualmente es desarrollado por un grupo de programadores sin ánimo de lucro. Aceleradamente se ha convertido en el más usado de los servidores en sistemas UNIX e incluso rivalizando con otros de plataforma Win32 ya que aunque inicialmente no era así.

Apache corre actualmente bajo casi cualquier plataforma: Linux, FreeBSD, OpenBSD, NetBSD, Amiga Os 3.x, Mac Os X, SunOs, Solaris, IRIX, HPUNIX, Digital Unix, UnixWare, AIX, SCO, ReliantUNIX, DGUX, OpenStep/Mach, DYNIX/ptx, BeOS y Windows y además su código es abierto y disponible para descarga sin ningún coste. Los desarrolladores de Apache se han esforzado en lograr escalabilidad, rapidez, flexibilidad y eficiencia, convirtiéndolo en el servidor más usado de la web (aproximadamente en el 60% de los sitios Web), según encuesta de Netcraft (www.netcraft.co.uk).

Arquitectura de Apache

Apache está estructurado en módulos que están escritos en lenguajes C o Perl y debido a que es de fuente abierta cualquiera puede escribir un módulo y publicarlo, por lo que existen gran cantidad de módulos de Apache disponibles en la Web. La configuración de cada módulo se hace mediante la configuración de las directivas contenidas en cada uno de estos. Pueden clasificarse en tres categorías:

- Módulos base: contiene las funciones básicas de Apache
- Módulos multiproceso: responsable de la unión con los puertos de la aceptando peticiones y enviando a los procesos hijos atender las peticiones.
- Módulo adicional: cualquier módulo que le añada funcionalidad al servidor.

Las funcionalidades más elementales se encuentran en el módulo base, siendo necesario un módulo multiproceso para manejar las peticiones.





Se han diseñado varios módulos multiproceso para cada uno de los sistemas operativos sobre los que se ejecuta el Apache, optimizando el rendimiento y rapidez del código. El resto de funcionalidades del servidor se consiguen por medio de módulos adicionales que se pueden cargar. Para añadir un conjunto de utilidades al servidor, simplemente hay que añadirle un módulo, de forma que no es necesario volver a instalar el software.

Configuración

Los ficheros de configuración de Apache se buscan por defecto dentro del directorio "/usr/local/apache/conf" aunque esto puede cambiar si se desea. Allí se encontrarán, si la instalación fue exitosa, los ficheros:

- httpd.conf: fichero principal de configuración de Apache.
- srm.conf: fichero de definición del espacio de nombres que los usuarios ven del servidor de Web. En este fichero también se especifica donde se encuentran los cgi-bin, los iconos, el tipo de documento por defecto, como se responde ante los errores, que fichero es el índice dentro de un directorio, donde está la página personal de los usuarios del sistema.
- access.conf: fichero de control de acceso global a los datos del servidor de Web. En él se especifica los permisos de accesos a directorios, ficheros y URLs dentro del servidor, así como diferentes configuraciones.
- mime.types: fichero de control de los tipos MIME que son enviados al cliente en función de la extensión del fichero. El fichero principal es httpd.conf, muy fácil de editar ya que se encuentra en formato texto y puede modificarse con sólo cambiar el estado de las directivas.

Características

Entre las principales características de Apache se pueden contar:

- Multiplataforma y código abierto: la primera y más importante es lo que hace a este servidor tan potente, pues está en constante actualización.
- Extensible: Se han desarrollado diversas extensiones entre las que destaca PHP y Perl.
- Negociación de contenidos.
- Directorios de alias
- Reescritura de URLs.
- Informe de errores HTTP configurable.
- Soporte de último protocolo HTTP 1.1
- Soporte para CGI (Common Gateway Interface): a través de los módulos mod_cgi y mod_cgid.
- Soporte de host virtuales: es uno de los primeros en soportar host virtual, tanto basado en direcciones IP (Una dirección IP para cada sitio Web) como basado en nombres (Más de un sitio Web con una sola dirección IP).
- Soporte de autenticación HTTP.
- Servidor Proxy integrado.
- Estado del servidor y adaptador de registros: Apache brinda una monitorización y reporte de estado del servidor.

Una de las características más destacadas en Apache es que puede escuchar las peticiones para diferentes direcciones IP. Con ello, se puede centralizar el servicio de diferentes dominios web con direcciones IP diferentes, en un único servidor de Web (hosting de varios dominios). En GNU/Linux esta característica se puede implementar compilando el núcleo con soporte para alias IP, lo que permite definir en una única tarjeta de red, varias direcciones IP.





Pueden encontrarse otros servidores en la web tanto libres como privativos. Algunos son:

- CERN httpd
- IIS
- Resin
- Tomcat
- Geronimo
- JBoss
- JOnAS
- Cherokee

El más usado de estos es el Internet Information Server (IIS) de la Microsoft Software Corporation y que viene incluido en el paquete de instalación de Windows. La polémica sobre cuál de los dos servidores es mejor es tan amplia como la seguridad Linux o Windows o cuál de los dos es mejor. No obstante veremos las principales características del IIS y algunas encuestas por Google y Netcraft para arribar a conclusiones.

IIS:

Ventajas:

- Es fácil de usar
- Posee interfaz gráfica (GUI) para la administración de servidores
- ASP preparado en la instalación por defecto

Desventajas:

- Multitud de fallos de seguridad
- Sólo funciona para Windows NT/2000 o superior
- La mayoría de la funcionalidad extra debe ser comprada separadamente.

Entre las desventajas que podrían señalársele Apache una es que toma tiempo de adiestrarse en su uso, no obstante cuenta con mucha documentación y manuales online. En opinión del autor la principal ventaja de Apache es que es un software de fuente abierta lo que posibilita que miles de programadores estén actualizando y tapando parches constantemente.

Por otra parte IIS es un producto estilo Microsoft. Provee interfaz gráfica de ventana e integración con otros mecanismos de Windows como Active Directory, MS SQL Server y los servicios Windows Security que en contraste lo hacen dependiente de estas aplicaciones (Active Directory), lo que puede constituir una limitante para las empresas ya que aumenta el costo.

Además de interfaz gráfica IIS también proporciona la capacidad de administrar uno o más servidores IIS de un navegador Web y algunas funciones de administración disponibles en otros productos de Microsoft, como un asistente de configuración y configuración y mantenimiento remotos.

Permite además conexión a cualquier base de datos ODBC e incluso más de un sitio Web puede ser conectado a una misma base de datos ODBC. A continuación se muestra una gráfica que ilustra el uso de varios servidores en la red. La línea azul es el crecimiento Apache y la roja el de IIS desde octubre de 2005 hasta Enero de 2008 (tomado de Netcraft).

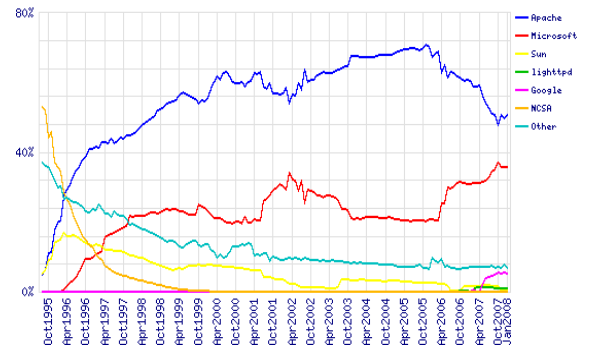
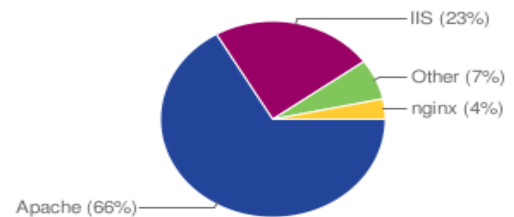


Figura 1 Gráfica de una encuesta de Netcraft

Las estadísticas siguientes fueron obtenidas del blog de Seguridad Online de Google. Lo hecho por Google es coger 80 millones de dominios de Internet, y determinar el software que está corriendo para albergar las páginas en cada caso. El resultado es el siguiente:



2 GráficaFigura de un estudio de Google

Estos números difieren respecto a los de Netcraft ligeramente, a causa de que Google sólo realiza para este tipo de análisis exámenes de URLs raíz, del tipo /index.htm, con lo que en ausencia de éstos, no se han contabilizado los resultados, cosa que sí hace Netcraft.

El estudio de Google era orientado a como se distribuye el malware a través del planeta en función de las soluciones más empleadas de alojamiento Web: IIS y Apache.



Escoger entre estos servidores puede conllevar a una discusión profunda en algunos aspectos, pues si usa IIS indudablemente está usando Windows o si usa Apache en Windows (análisis aparte) o en Linux. En esta última, Apache se convierte en una poderosa herramienta y forma parte de la solución LAMP (Linux, Apache, MySQL y PHP), de las más usadas en la actualidad.

Apache es hoy si no el principal, al menos uno de los más usados entre los servidores de Web. Por su extensibilidad y naturaleza de código abierto es muy eficiente para instalar en máquinas GNU/Linux ya que asegura así, buenas comunicaciones en un entorno seguro. Apache ha hecho que GNU/Linux posea un campo más amplio en el mundo de Internet y se ha convertido en una plataforma de gran potencia difícil de superar con muchísimos seguidores en todo el mundo.

Referencias

<http://www.networkcomputing.com/>

<http://www.apache-php-mysql.com/>

Alvaro del Castillo San Felix. El servidor de web Apache: Introducción práctica Apache 1.x y 2.0 alpha(ftp://ftp.prod.uci.cu/Disenno_web/docum/Books/apache.pdf)

Ben Laurie y Peter Laurie. Apache. The definitive guide. Estados Unidos. O'Reilly & Associates, Inc., 1999

HUMOR LIBRE





Marisniulkis Lescaille Cos
mlescaille@estudiantes.uci.cu
Facultad 2



EVENTOS

Festival de Software Libre hasta la Lenin

La tarea inicial más que tarea constituía un reto: extender el Festival de Software Libre a los Centros de Enseñanza Media, fundamentalmente los Institutos Politécnicos de Informática (IPIs) toda la vorágine que a favor del software libre se viene desarrollando en la facultad 2, así llegó el festival hasta el politécnico Julio A. Mella en el municipio Boyeros de esta ciudad capital.

Tan sólo meses después una nueva misión vería la luz: el Instituto Politécnico Vocacional de Ciencias Exactas (IPVCE) Vladimir I. Lenin, reconocido en todo el país por su compleja estructura, los resultados obtenidos en diferentes eventos y por ser el más grande.

Para el grupo de estudiantes que protagonizaron el naciente festival en el pasillo del docente 1 esto realmente constituiría un desafío. El Julio Antonio había sido una prueba, pero este sería como la temida de Nivel. Pues para un grupo creado hace poco más de un año, tantas actividades de tal envergadura constituían una muestra de que el trabajo no fue en vano y que continuaría.

Hechas las coordinaciones con la dirección del centro estudiantil el lunes 4 del presente mes partieron hacia la escuela, no sin contratiempos, pues hasta la merienda inventó prepararse a la hora de salida. Y estos no concluyeron porque muy cierto es que no se logró mantener ese primer día encendidas las 20 PCs al mismo tiempo, a pesar del esfuerzo del electricista y de muchachos como Álvaro y Javier, devenidos multitareas.



El primer encuentro con los estudiantes de la Lenin, nos dejó bastante consternados. No parecían ni por mucho interesados en nuestra propuesta, pero la realidad demostró la validez de aquel refrán popular: Ver para creer. Durante los días subsiguientes el interés general aumentó considerablemente y pudo observarse cómo se montaban y desmontaban todas las teorías que rondaban en aquellos de la universidad más joven de Cuba, relacionadas fundamentalmente con la ubicación laboral, el ingreso, situación material de la beca, claustro de profesores, entre otras.

Es de resaltar además que la asistencia de varones fue superior por alto rango que la de las hembras y los primeros por lo general se mostraban más motivados en el tema, lo que puede dar al traste con que cada año las cifras de entrada a la universidad se alejen más unas de otras. Curioso, fueron muchachas las que más investigaban lo concerniente a qué es la programación, e incluso convidaron a algunos del grupo a enseñarlas a programar (tarea difícil).

Entre las áreas de exposición más visitadas estuvieron la de juegos, diseño, programación, y los proyectos GPI, NOVA y la revista UXI que han colaborado con el equipo que viene desarrollando estos festivales, y el proyecto Procyon. Aunque también se presentaron muestras de aplicaciones de Bases de Datos, Portal de la Cátedra de Programación Avanzada y de varias distribuciones GNU/Linux.



Las actividades comenzaban desde la mañana con conferencias-debates en las que estuvieron al frente José Ramón Sera, estudiante de cuarto año de la facultad 10 y Edgar González presidente de la FEU del panal (Facultad 2), así como el Ing. Abel Meneses Abad, líder del proyecto Unicornios, ya asiduo en el apoyo a las actividades que desarrolla el grupo; y se extendieron durante toda la semana, en un encuentro fructífero, tal como asegura José Ramón:

“Se ha observado un interés cada vez mayor. No puede culminar aquí, debe tener una segunda fase en la UCI, para que los muchachos interactúen con el ambiente de allí y vean cómo funciona todo.”

Por su parte dos estudiantes de este centro ofrecieron sus impresiones. Olivia:

“Ha sido una experiencia muy bonita, hemos podido aprender cosas que no teníamos ni idea, además esto aligera la carga de la escuela, es una opción para pasar el tiempo libre”,



Dana afirma que les interesa mucho la UCI, “quisiera conocerla”, dice. Agrega que es impresionante como cada año en la universidad se van graduando jóvenes con alto nivel de aprendizaje en la informática y se los transmiten a los demás, “pues nosotras aprendimos de GNU/Linux y del software libre, que no sabíamos nada, y además pudimos sentarnos en las máquinas e interactuar con él. De veras fue una experiencia enriquecedora.”

Pero para muchos, en especial para Sandy Noa, el jueves constituyó el día más especial de la semana: luego del esfuerzo desplegado durante toda la mañana el primer mensaje a través del chat llegaba a una PC vecina, y vale mencionar que era de mensajería instantánea pues ya se contaba con un pequeño servicio de correo, pero lo valedero de este mensaje no fue ni su contenido(“siiiii”) o lo novedoso, pues el chat está inventado hace ya un tiempo, sino porque Sandy había aprendido la noche anterior cómo instalar este servicio y a costo de todas las horas de la primera mitad del día logró ponerlo en práctica; la alegría se hizo general y en un momento había todo un diálogo en la improvisada red.

El último día de presentaciones cerró con actividades bailables concurso incluido, previo se desarrolló un encuentro de conocimientos, donde participaron estudiantes de décimo y oncenno grados, dándole conclusión a un ya histórico encuentro, que muchos esperan se repita.



**Movimiento de SWL
Facultad 2**





INFORMÁTICA 2007



Karel Reynaldo Piñero
karel@lt.desoft.cu
DESOFT
Cuba

Manuel Alejandro Gil Martín
chony@uci.cu
Universidad de las Ciencias Informáticas
Cuba

“OPEN SOURCE BUSINESS INTELLIGENCE”

La inteligencia de negocio (BI) consiste en generar la información necesaria para tomar decisiones oportunamente, a partir de datos provenientes de múltiples fuentes. Por el mercado a que están dirigidas, las herramientas propietarias para BI se caracterizan por sus altos precios y licencias restrictivas, con lo que quedan inaccesibles a un amplio sector que pudiera beneficiarse significativamente de las mismas, pero no puede asumir el costo asociado a la adquisición de una solución de este tipo.

Este artículo presenta la posibilidad de extender la inteligencia de Negocio a las más diversas organizaciones, utilizando para ello herramientas gratuitas disponibles en la Comunidad Open Source.

Se resumen las características e inconvenientes de las soluciones propietarias tradicionales, y los beneficios asociados a la adopción de una solución abierta. Queda demostrada, finalmente, la fortaleza de la Comunidad Open Source y su capacidad para competir en igualdad de condiciones con empresas de reconocido prestigio a nivel internacional, quienes destinan cuantiosos recursos para desarrollar y mantener sus productos y servicios.



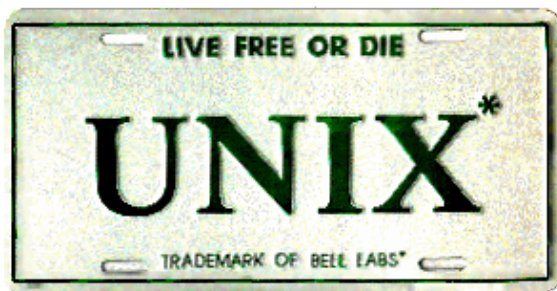
“PROCESO DE DESARROLLO DE APLICACIONES DE ACUERDO A LOS CONCEPTOS DE SOFTWARE LIBRE”

En el Software Libre se aboga por un modelo de desarrollo que gira en torno a la comunidad. Existen innumerables ventajas al desarrollar un producto aprovechando las capacidades de trabajo y conocimiento de personas con los mismos intereses, que colaboran para lograr un objetivo común. Sin embargo, este desarrollo está sujeto a ciertas reglas que garantizan que el trabajo se haga con la calidad requerida y en una manera organizada.

En este trabajo se exponen los pasos necesarios para crear un entorno colaborativo en una institución, a su vez, se muestra la experiencia alcanzada por la Universidad de las Ciencias Informáticas en este sentido y los resultados obtenidos.

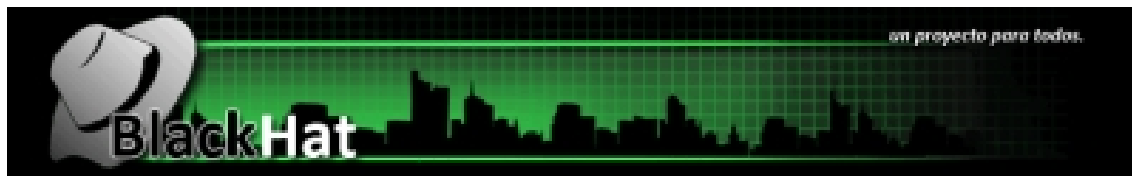
BSD

http://www.freebsd.org/doc/es_ES.ISO8859-1/articles/explaining-bsd/



Ponencias disponibles en:

http://10.128.50.121/Documentacion/Publicaciones/06_Febrero_2
(Sólo disponibles desde la UCI)



BlackHat es un proyecto de divulgación científico-técnica para todos los interesados en la Informática y las ciencias afines, implementado con el objetivo de elevar la cultura informática en Cuba mediante la creación de una comunidad de profesionales y estudiantes involucradas en el tema

Para suscribirse enviar un correo a:

blackhat4all@gmail.com

con Asunto: Suscripción

OpenOffice 3.0

La versión más importante y esperada de la Suite Ofimática Libre



Beta: 30 de Abril de 2008

RC1: 25 de julio 2008

Versión Final: Septiembre 2008



Voicing the Community
OpenOffice.org 3.0 and Beyond