



UXI.deb

Revista cubana de Software Libre



v. 1.0-uci

Compendio de artículos del Volumen I

```
$ ./configure --prefix=/usr --mandir=\${prefix}/share/man --infodir=\${prefix}/share/info
```

```
$ make install DESTDIR=`pwd`/tmp
```

Source: UXI

Section: Revistas de SWL

Priority: optional

Maintainer: Universidad de las Ciencias Info

Build-Depends: kpdf >= 0.5.9

Standards-Version: 1.0

Package:

Architecture: any

Depends: \${shlibs:Depends}

Description:



Publicar nuestro software: único camino hacia la libertad plena.



Edición:

Abel García Vitier
avitier@estudiantes.uci.cu

Revisión y Corrección:

Felix Daniel Batista Diñeiro
fdbatista@estudiantes.uci.cu

Arte y Diseño:

David Padrón Álvarez
dpadron@estudiantes.uci.cu

Patrocinado por:

**PROYECTO
UNICORNIOS**



Al Lector

Este material constituye una recopilación de todos los artículos publicados en el primer volumen de la Revista Cubana de Software Libre: UXi. A petición de la comunidad, seleccionamos aquellos artículos que fueron publicados en los primeros diez números, durante el año 2007. El objetivo primordial es facilitar la consulta de los mismos sin necesidad de buscar en cada número.

En el índice podrá encontrar los títulos de los referidos materiales, los que se encuentran ordenados por fecha de publicación. Muchos de ellos quizás carezcan de actualidad teniendo en cuenta la fecha en que fueron publicados; sin embargo, consideramos que aún tienen mucho que aportar a aquellos lectores interesados en la temática que se aborda. Otros, por su parte, trascienden todavía y pueden ser de gran utilidad.

A medida que usted avance en la lectura podrá apreciar lo que se ha llamado “evolución de UXi”, lo cual se aprecia en la ampliación de las temáticas abordadas, siempre dirigidas a un público partidario del Software Libre.

En estos momentos también se está trabajando en un segundo compendio de artículos correspondientes al segundo volumen, que culminará en marzo del 2009 con un compendio de todo cuanto fue y será publicado en el presente año, y que sin dudas será superior, tanto en cantidad como calidad de los artículos, a este que hoy sometemos a su consideración.

De esta forma, sólo nos resta desearle una buena lectura e invitarlo a que contribuya también a socializar el conocimiento.

Licencia:

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>



Índice

Al lector.....	4
Software Libre para una sociedad libre: GNU/Linux, un futuro para Cuba.....	5
Videos de Inter-Nos en GNU/Linux.....	8
Evolution-config-exchange.....	9
Hay que migrar al Software Libre.....	10
Sistema Adaptativo de Filtrado de Contenidos.....	13
Riesgos de seguridad utilizando system() o popen().....	24
¿Qué proyectos usan y desarrollan Software Libre en la UCI?.....	26
Tocororo GNU/Linux.....	29
Infodrez, el sitio Web del ajedrez de la UCI.....	32
Autenticación de Plone contra el servidor de LDAP de la UCI.....	35
Creación de un repositorio local.....	36
Las patentes de software, un problema legal.....	37
NOVA GNU/Linux.....	39
Metodología para la migración a Software Libre (Parte I).....	42
Entorno de Desarrollo para Python.....	44
Concurso de Software Libre de la UCI.....	47
Tiflotecnología: accesibilidad en GNU/Linux.....	49
Subversion y Apache 2.....	51
Perl en sólo una línea.....	54
Optimizar el arranque en sistemas GNU/Linux.....	56
LinuXierra, primera comunidad de Software Libre en las montañas cubanas.....	59
Diseñar con Blender en GNU/Linux.....	63
Una vía para la enseñanza de GNU/Linux en los Joven Club	66
PHP5 vs PHP4.....	68
How To GDM	72



<u>GNU/Linux en los CDI: Una experiencia libre.....</u>	<u>77</u>
<u>“Los momentos más importantes del SWL en nuestro país son los que estamos viviendo”</u>	<u>80</u>
<u>Construyendo una terminal</u>	<u>82</u>
<u>Sistema de Almacenamiento Distribuido</u>	<u>85</u>
<u>How To: Configurar Fluxbox</u>	<u>91</u>
<u>David contra Goliat: Migración de la MiniUCI de Granma.....</u>	<u>93</u>
<u>Cómo crear y usar una librería estática.....</u>	<u>96</u>
<u>Creando un tema de boot splash.....</u>	<u>101</u>
<u>Cambiar apariencia de servidor Apache.....</u>	<u>103</u>
<u>El Software Libre y los procesos pedagógicos contemporáneos.....</u>	<u>105</u>
<u>Servicios telemáticos sobre GNU/Linux.....</u>	<u>110</u>
<u>Servicio de Streaming: video en demanda usando VLC.....</u>	<u>112</u>
<u>Solución integrada de la implantación de un sistema de perfiles centralizados.....</u>	<u>113</u>
<u>GIMPmanía.....</u>	<u>118</u>
<u>Creación y uso de librerías dinámicas.....</u>	<u>120</u>
<u>Programación de socket en Linux con C++.....</u>	<u>126</u>
<u>Encriptar el sistema de ficheros en Debian & Ubuntu.....</u>	<u>132</u>
<u>El buscador, la solución contra el tiempo.....</u>	<u>135</u>
<u>Aplicación de la estrategia de migración a Software Libre.....</u>	<u>137</u>



Software Libre para una sociedad libre: GNU/Linux, un futuro para Cuba

Michel Hernández Rodríguez, Ing. Abel Meneses Abad
mhrodriguez@estudiantes.uci.cu, abelma@uci.cu
Universidad de las Ciencias Informáticas

GNU/Linux es una familia de sistemas operativos desarrollada por comunidades libres y empresariales que en los últimos años se ha alzado como una alternativa ante los sistemas propietarios imperantes en el mercado mundial.

En la actualidad, varios países, incluido el nuestro, han iniciado un proceso de cambio y desarrollo de este sistema operativo, para así contribuir con la progresiva eliminación del software propietario. Este proceso de migración de software incluye un análisis de las soluciones existentes y la búsqueda de aplicaciones equivalentes en el área de software libre para aportarle nuevas características a dicho sistema, así como diferentes comodidades y facilidades para el uso de los usuarios finales.

¿Por qué la migración a SWL?

En la actualidad nuestro país hace un uso bastante elevado del software propietario, fundamentalmente del sistema operativo Windows y demás programas que suelen acompañarlo, en su inmensa mayoría también propietarios. Ahora, como bien se conoce, cada aplicación de este tipo tiene asociado un precio o costo inicial, sin mencionar el precio adicional asociado al mantenimiento de los mismos en caso de ser necesario, o simplemente el valor de otras aplicaciones quizás necesarias para un mejor rendimiento o eficiencia del programa inicial. Y estamos analizando esta situación de forma independiente, porque si una misma aplicación es usada en diez computadores diferentes, su precio sería diez veces mayor, ya que para el uso corporativo se toma en cuenta cada distribución del programa por separado, y se cobra por cada PC donde sea instalado. Resulta este un proceso muy costoso, incluso para los países más desarrollados.

Por otro lado, cada día que pasa, los principales monopolios de las telecomunicaciones y la informática siguen afianzándose en países del primer mundo, lo cual incrementa la dependencia de los países más pobres. Y mirando con mucha más perspectiva la situación, no solo sería de gran utilidad el uso del Software Libre para Cuba, sino su producción, y con ello la posibilidad de establecer convenios con otros países para la comercialización de estas distribuciones o aplicaciones soportadas en GNU/Linux. De esta forma, el comprador o cliente necesitaría hacer uso de nuestro soporte y servicios técnicos, generando una fuente de ingresos para la mayor de las Antillas.

Algunas ventajas y motivos.

Desde hace algunos años en nuestro país se vienen desarrollando algunas distribuciones de Linux tales como NOVA (en nuestra universidad), Caimán, Linux UH y variados programas incluidos en ellas para facilitar su uso, con el objetivo de llevar a cabo un proceso de migración hacia el software libre en el país debido a las múltiples ventajas y facilidades que proporciona, algunas de las cuales veremos a continuación.

1. **Todo software libre está bajo los efectos de algunas licencias como la GPL:** esta licencia está orientada a proteger y garantizar la libre distribución, modificación y uso de los programas que ampara. Su propósito es declarar que el software cubierto por esta licencia es libre, y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios. Ello permite modificar una copia o copias del programa o cualquier porción de éste, y poder copiar y distribuir esa modificación o trabajo bajo los mismos términos de la GPL.

Nota: Cuando hablamos de software libre, estamos refiriéndonos a libertad, no a precio.



Solamente con esta ventaja, el país ahorraría el pago de las decenas de miles de dólares que tendría que abonar en caso de que se eliminara el bloqueo contra Cuba, y a continuación lo veremos con el siguiente ejemplo:

A finales del 2006 el país contaba con un potencial tecnológico de 6000 computadoras asignadas solamente a los Joven Club de todos los municipios del país. Si cada una de estas computadoras tiene instalado software propietario, esto significaría que esta cantidad de equipos representaría una media de 90 dólares por cada versión de Windows XP en cada PC. Ello sumaría un total de 540 mil dólares como mínimo, solamente tomando en cuenta el precio del sistema operativo. Una media de 112 dólares mínimos por cada distribución del paquete de Office 2003 representaría 672 mil dólares. Y ya estaríamos hablando de 1.212 millones invertidos en el software básico para estos equipos, sin analizar el costo de muchas otras aplicaciones que dependen del uso que se haga del computador, ni de los programas o controladores necesarios para los diferentes periféricos o dispositivos conectados, lo cual se adicionaría también a estos más de 1.2 millones de dólares.

Ahora, si a todos estos equipos de cómputo se les instalara una distribución de Linux, la mayor parte de este gasto -podría ser incluso toda la suma- se le evitaría al país. ¿Cuánto se ahorraría si se hiciera esto en las más de 300 mil computadoras que tiene el país?

2. **Es más seguro:** la gran mayoría de los ataques de hackers son dirigidos a servidores Windows, así como los virus, los cuales se centran principalmente en ése sistema operativo. Además, la plataforma Linux es más robusta, lo cual dificulta los intentos de violar el sistema de seguridad de Linux.
3. **Más rápido:** al tener una plataforma más estable, se favorece el desempeño de aplicaciones de todo tipo, tales como bases de datos, aplicaciones XML, multimedia, etc. La eficiencia de su código fuente hace que la velocidad de las aplicaciones Linux sean superiores a las que corren sobre Windows.
4. **Más económico:** requieren menos mantenimiento. En servidores Windows es más costoso debido a que es necesaria una frecuente atención y monitoreo contra ataques de virus, hackers y errores de código, instalación y actualización de parches y service packs. El software libre, así como un sinnúmero de aplicaciones de código abierto, no requiere supervisión tan estrecha ni pagos de pólizas de mantenimiento necesarias para obtener las actualizaciones.
5. **Flexibilidad y capacidad de adaptación del código abierto:** una de las principales ventajas del código abierto es su gran capacidad de adaptación. A nivel de kernel, éste se puede optimizar para que proporcione un mayor rendimiento como sistema de red. A su vez, las utilidades del sistema operativo pueden ser recompiladas para hacer un mejor uso del procesador, y se pueden eliminar servicios innecesarios para aumentar su productividad. A su vez, tener una gran comunidad de desarrolladores de GNU/Linux es una garantía de seguridad de que este sistema seguirá expandiéndose y convirtiéndose cada día en una plataforma más potente. La libertad de poder modificar el código fuente es muy valorada por los usuarios, pues permite que ellos mismos hagan cambios y configuraciones a sus aplicaciones, sin necesidad de contar con nadie ni requerir de licencias o permisos para ello.
6. **Es austero:** GNU/Linux funciona hasta en un PC 386 con algunas de sus mejores ventajas, aplicaciones y desarrollos más actuales.
7. **Gran variedad:** hay miles de programas libres para Linux, adaptados a muy diversos propósitos y disponibles en internet. Además, existen muchos programas propietarios que corren en este sistema, como Zend Studio, Visual Paradigm, MatLab, etc.
8. **Facilidades:** aunque no como Windows, GNU/Linux ya no está restringido a personas con grandes conocimientos de informática; sus desarrolladores han hecho un gran esfuerzo por dotar al sistema de



asistentes de configuración y ayuda, además de un sistema gráfico muy potente. Distribuciones como RedHat / Fedora tienen aplicaciones de configuración similares a las de Windows.

Conclusiones:

Como se ha podido constatar, el software libre brindaría mayor seguridad a quienes hagan lo utilicen y a nuestra sociedad en general. La robustez, flexibilidad de configuración, capacidad de adaptación del código abierto, seguridad y rapidez, entre otras ventajas, son factores de gran peso a la hora de hacer una valoración de GNU/Linux.

No siempre debe primar el criterio de que Windows es más cómodo, y por tanto la mejor solución para nuestro país. No obstante, las fuerzas tendrían que estar divididas entre ambos sistemas, porque de esa forma se garantizaría el conocimiento de ambos y no se verían las cosas de manera parcial. Ello permitiría, además del conocimiento de ambos sistemas, que cualquier empresa productora de software esté preparada para satisfacer solicitudes de sus clientes sobre cualquiera de las dos plataformas.



Videos de Inter-Nos en GNU/Linux

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

Algunos años después de creada la Universidad de las Ciencias Informáticas, se comenzó a dar los primeros pasos en el uso del sistema operativo GNU/Linux; al poco tiempo se habían aprendido muchas cosas, logrado compatibilizar muchas aplicaciones y hacer casi todo lo que se hacía con Windows, pero siempre quedaron problemas pendientes. Uno de ellos fue cómo visualizar los videos en los reproductores libres de GNU/Linux ubicados en un servidor streaming (mms) como el de Inter-Nos. Finalmente se logró ver los videos del mms con la aparición y descubrimiento, para los navegadores libres, de la cadena *network.protocol-handler.app.mms*, que al agregarlo al navegador permite al usuario elegir el reproductor con el que desea ver los videos, pero el problema no estaba resuelto aún.

Microsoft Media Server es el servidor de media de Microsoft, servidor de flujo que brinda su servicio en tiempo real y compatible solamente con reproductores propietarios como Windows Media Player. ¿Qué problemas trae esto para los reproductores libres? La respuesta es sencilla y la solución compleja: no se pueden correr los videos de Inter-Nos desde GNU/Linux. Esto trae consigo un nuevo problema que se vuelve aún mayor cuando el 25 de marzo del 2006 la facultad 10 migra su infraestructura docente completamente a GNU/Linux, pues los estudiantes presentaban problemas al tratar de acceder a teleclases y videos.

Surge la solución

El pasado año, luego de un largo estudio investigativo del problema, se llegó a la solución: un sistema desarrollado en el lenguaje de Harry Wall con un conjunto de scripts desarrollados en PERL, cuyo principio fundamental se basa en hacer un búfer local en el sistema donde se va guardando el video y a la vez se va reproduciendo. Cuando se pulsa sobre el link del video en la página Web, de manera automática y transparente a los usuarios, se comienza a descargar el video y a reproducir; solo se le muestra una pequeña ventana en la parte izquierda del monitor que les informa a los usuarios el nombre del video en el servidor mms y el tiempo inicial para bufferear, este último lo configura el usuario.

El sistema está disponible en nuestro servidor de documentación¹ con una versión y un instalador elaborado específicamente para la imagen de los laboratorios de la facultad 10. Actualmente han surgido nuevas soluciones de mayor aceptación, pero existen aun algunos problemas para ciertos formatos de video, como es el caso del reproductor VLC disponible para las últimas versiones de los sistemas GNU/Linux en todos los repositorios de la UCI. No obstante, para aquellos usuarios que usan sistemas estables, la solución explicada en este trabajo sigue siendo la más útil, y en muchos casos se hace necesaria, pues VLC no cubre todos los formatos necesarios.

Todos los usuarios de la comunidad pueden contribuir al desarrollo de esta modesta solución. Más información sobre este sistema está disponible en la documentación, donde se explican sus características de manera íntegra².

¹ <http://10.128.50.121/Software/Instaladores/Generales/Internos>

² http://10.128.50.121/Documentacion/Migracion/Interno_X/



Evolution-config-exchange

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

Se ha desarrollado un programa para configurar las cuentas de correo electrónico automáticamente y posibilitar el uso del cliente de correo Evolution para todo el que llegue a la computadora y quiera revisar su correo electrónico. La aplicación está pensada para la imagen de la Facultad X, pero hay que realizarle algunas modificaciones para este propósito; no obstante, ya está disponible para usuarios individuales.

Esto funciona para los usuarios de Debian testing/unstable, pues las librerías de KDE para Ubuntu Edgy son versiones menores que las que se utilizaron en Debian. Esperamos resolver este inconveniente pronto para que los usuarios de Ubuntu puedan instalarlo también.

Además, esta es la primera gran prueba de la aplicación, así que le pedimos a todos los usuarios que la utilicen y envíen sus apreciaciones, opiniones, o sugerencias.

Detalles de la Aplicación:

Nombre: evolution-config-exchange
Versión: 1.1
Lenguajes utilizados: C++, Bash
IDE: KDevelop

Como no está en el repositorio, deben tener instalado los siguientes paquetes para evitar errores de dependencias:

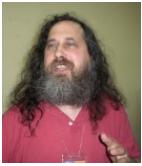
evolution (>=2.6.3-3)
evolution-exchange (>=2.6.3.dfsg-1)
libqt3-mt (>=3:3.3.7-2)

Recomendado: kdelibs(>=4:3.5.5a.dfsg.1-5)

Para instalar el paquete, en la consola ejecutar:

```
dpkg -i /ruta/evolution-config-exchange_1.1_i386.deb
```

Disponible a partir de : Lunes 29/01/2007 8:00am
Lugar de descarga: [http://10.33.20.167:3389 /package](http://10.33.20.167:3389/package)
Archivo a descargar: evolution-config-exchange_1.1_i386.deb



Hay que migrar al Software Libre

*Entrevista a Richard Stallman
Por Juventud Rebelde
Febrero 2007*

Una de las personalidades más controvertidas de las nuevas tecnologías de la Información y las Comunicaciones a nivel mundial, fundador del movimiento del software libre y enemigo público de Microsoft, serían credenciales más que suficientes para Richard Stallman. Como si no bastara, a todo ello une una personalidad muy singular, que lo lleva a gustar de “reír, incluso de mí mismo” o a escribir una parodia de la Guantánamera, titulada Guantánamero, para criticar la prisión que Estados Unidos mantiene en la ilegal Base Naval.

Nacido el 16 de marzo de 1953 en Manhattan, Nueva York, se autodenomina «Santo Ignucio», en referencia al código GNU que él mismo escribió y es una de las bases del software libre. Confiesa además que siempre quizo ser comediante y gusta de terminar sus charlas, no importa el lugar donde esté, disfrazado con un viejo disco rígido como aureola y un manto improvisado, “bendiciendo” las computadoras para que decidan migrar a software libre.

Su propuesta de que el software sea de uso libre, en la más amplia acepción del término, le ha traído innumerables enemigos. Pero no desiste. Anda de un lado para otro por el mundo defendiendo sus puntos de vista. Esas mismas ideas e iguales propósitos lo han traído ahora, por primera vez, a Cuba, para asistir al III Taller Internacional de Software Libre, en el contexto de la XII Convención y Exposición Internacional Informática 2007, en el Palacio de Convenciones de La Habana.

Solo puso dos condiciones para dialogar en exclusiva con JR, durante más de una hora y media, en su habitación del Hotel Palco. La primera, que no se utilizara el término abierto como “libre”, y la otra, que se hablara del sistema GNU/Linux, y no solo de Linux, “que es el nombre de un componente que escribió LinusTorvald. Cuando se habla de este, es justo que se diga Linux, pero cuando se trate del sistema se debe decir GNU/Linux. Yo solo quiero modestamente la mitad del reconocimiento”.

La entrevista a Stallman, muchas veces criticado por sus “excentricidades”, como no usar teléfono celular o andar con una vieja laptop en la que utiliza un sistema “que no contiene ningún software privativo”, fue en realidad una conversación con una personalidad contradictoria pero afable, que gusta de los términos precisos y defiende sus ideas, aunque muchas veces se le señale por sus puntos de vista.

“Eso no me molesta, asegura. Muchos no están de acuerdo con el software libre y les es útil atacarme a mí, porque no pueden hacerlo con la filosofía. Tienen el derecho. No me incomoda. Eso sí, me disgusta que en muchos artículos, por ejemplo, me llamen evangelista del código abierto, porque no apoyo esa filosofía. No apoyo solo el código abierto, o sea que la gente nada más conozca cómo funciona un programa, sino el software libre. Y la diferencia es que para nosotros es un asunto ético. El software privativo no es ético porque no respeta la libertad, y el código abierto no se interesa por esta cuestión. Si yo soy el “padre del código abierto” se hizo con “esperma” que me robaron sin mi consentimiento”.

Libertades Importantes

Stallman aclara que para los defensores del software libre la cuestión no es sólo que el código de un programa sea abierto, que se sepa cómo funciona, sino que el usuario tenga las cuatro libertades fundamentales planteadas por el movimiento: “la libertad 0, la de ejecutar el programa como quieras; la libertad 1, estudiar el código fuente y cambiarlo para que el programa haga lo que quieras; la libertad 2, la de ayudar a tu prójimo, o sea distribuir copias del programa a los demás cuando quieras; y la libertad 3, la de ayudar a tu comunidad distribuyendo copias modificadas cuando quieras.”

“Ninguna de estas libertades es una obligación. Es decir, nunca el usuario está obligado a aplicar las cuatro cosas, pero tiene la libertad de hacer cualquiera cuando quiera”, explica.



El padre del Software Libre (SL) asegura que este movimiento se puede comparar con lo que sucede con las recetas de cocina. “Esta analogía es útil porque las dos son obras del conocimiento, y los cocineros cambian libremente las recetas. Sería ilógico que un día el Estado les diga que si cambian una receta le llamaremos pirata. Imaginen el enojo de los cocineros. Con ese mismo enojo fue que iniciamos el movimiento de software libre”.

Richard Stallman asegura que no le molesta ser un enemigo para Microsoft, que frecuentemente ataca y critica el código abierto, pero no menciona el software libre.

“Es porque su disputa con el código abierto es superficial. Los dos están de acuerdo en que su meta es el uso práctico, tener programas flexibles, usables, fiables, baratos, eficientes, y cada uno tiene su propia idea de cómo realizarlos.”

“Ahora, los defensores de esto no dicen que el código no abierto de Microsoft es inmoral. Nosotros sí lo afirmamos. Nuestra disputa con ellos es mucho más profunda, es una cuestión filosófica. Microsoft tiene cómo responderle a los del código abierto, aunque sea con argumentos usualmente falsos, pero no tiene manera de contestarnos a nosotros, porque no puede ni siquiera fingir que respeta la libertad del usuario o la proporciona”.

No me molestan las copias

Con una apretada agenda que incluyó la visita a la Universidad de las Ciencias Informáticas, además de otros eventos, Stallman asegura estar contra el bloqueo norteamericano a Cuba, y no molestarle, más bien inquietarle, que se copie ilegalmente software privativo para su uso.

“No me disgusta que se copie un programa en contra de la ley de otro país, porque lo que es ilegítimo es la ley, no la copia. Romper una ley injusta no es malo. Claro está -explica- estos programas no llegan a ser libres solo por poder copiarse, pues no cumplen con las otras libertades de las cuales ya hablamos. Sin estas, el desarrollador mantiene su poder sobre los usuarios al decidir qué hace o qué no hace el programa. Se necesitan las cuatro libertades para que el usuario tenga el control completo de su propia computación y pueda participar en una comunidad solidaria”.

Al respecto Stallman anunció que ya se trabaja en una tercera versión de la Licencia Pública General GNU GPL creada por él y que defiende la Free Software Foundation, para proteger la libre distribución, modificación y uso de software, y así evitar los intentos de apropiación que restrinjan esas libertades a los usuarios.

“Ya hemos publicado dos borradores y queremos publicar muy pronto el tercero, que será el final, aunque no sabemos cuándo estará definitivamente. Es un proceso difícil. Incluso, en la comunidad de SL la mayoría no presta atención a nuestras metas de libertad y solidaridad social, y muchas distribuciones de GNU/Linux incluyen programas privativos, lo cual es una debilidad.”

“Aunque a muchos no les guste, de cierta forma sí somos un movimiento político, un movimiento social por la libertad de los usuarios de computadoras y la solidaridad social. Es un movimiento político pero estrecho, sólo para un campo de la vida; lo que no implica que otros no tengan importancia.”

Igualmente señala que sigue con atención los proyectos de gobiernos como el de Venezuela, quien decretó una ley para que todas las instituciones del Estado migren a software libre, así como iniciativas en la India o en Extremadura, España, donde el gobierno de la región está migrando completamente y su reto es terminar este proceso para julio de este año.

“Y ha hecho la migración más importante, que es la de las escuelas. Estas tienen gran importancia porque son las que deben formar a los usuarios del futuro, habilitados para participar en una sociedad libre y solidaria, y eso sólo es posible enseñando el software libre”.



- No obstante , muchos afirman que prefieren sistemas operativos como Windows porque son más amistosos con el usuario.

- Era así hace años, cuando no teníamos entornos gráficos fáciles. Pero ahora ya no hay obstáculos, excepto la falta de acción. Incluso, suponiendo que sea mucho más sencillo de usar el software privativo, no cambiaría nada, porque es un asunto de libertad y solidaridad social.

Cuba puede

Stallman asegura que le gustaría que no solo Cuba, sino todo el mundo migrara por completo a software libre. “Para eso el mayor obstáculo es la inercia social. Pero Cuba tiene experiencia en luchar contra fuertes obstáculos. Entonces puede hacerlo”. Según su visión, el uso de SL contribuye al desarrollo en un concepto más amplio, “pues el software privativo es dependencia, y eso conlleva a la colonización electrónica. Las empresas de software privativo quieren colonizar a la gente de todos los países, y lo han hecho en Estados Unidos, Europa, en otros lugares del mundo, incluso en alguna medida en Cuba, excepto a los usuarios que usan el software libre”.

“La mayoría de los usuarios en el mundo todavía emplean el software privativo. Es triste pero nos demuestra que aún tenemos mucho por hacer. Espero que algún día esa situación cambie . Es mi esperanza, pero debe hacerse paso a paso. Y Cuba puede darlos”.

Libre vs. Privativo

El Software Libre es el que, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente . Suele estar disponible gratuitamente en, o al precio del costo de la distribución a través de otros medios. Sin embargo esto no es obligatorio y, aunque conserve su carácter de libre , puede ser vendido comercialmente. En cambio, el software privativo son aquellos en los que los usuarios tienen limitadas las posibilidades de usarlo, modificarlo o redistribuirlo, o no está disponible, o el acceso a este se encuentra restringido.





Sistema Adaptativo de Filtrado de Contenidos

Ing. Alain Guerrero Enamorado, Ing. Siovel Rodríguez Morales, Dovier Antonio Ripoll Mendez, Luis Enrique Sánchez Arce, Yordan Suarez Santiago, Jorge Luis Rodríguez Carpio, Jose Ramon Hermosilla Moreno, Kiuver Kaddiel Ibañez Castro, Oscar Andres Casas Machado, Saimel Saez Estrada, Israel Daniel De La Cruz Piñeiro, Ing. Joelsy Porven Rubier

alaing@uci.cu

Proyecto FILPACON



Introducción

Internet contiene una amplia variedad de contenidos, sin embargo, algunos pueden resultar ofensivos e incluso ilegales en muchos países. Hasta el momento, Internet por sí misma no es capaz de diferenciar los materiales de acuerdo a sus contenidos; su funcionamiento distribuido la convierte en una red en la cual las páginas son colocadas sin ningún tipo de revisión previa ni reserva. El problema fundamental de esto radica en que, de la misma forma en que contenidos de incuestionable valor educativo, científico, informativo, artístico, recreativo y muchos más son colocados en esta red, también son publicados contenidos en algunos casos de dudoso valor, y otros degradantes e incluso ilícitos. Cada día el número de sitios Web que albergan contenidos: pornográficos, pedofílicos, terroristas, xenofóbicos, racistas, extremistas, violentos y muchos más va en aumento.

La complejidad del problema se ve empeorada por la no jurisdicción de las leyes. Así, por ejemplo, de nada sirve que en Alemania, Francia, Austria e Italia tengan leyes que prohíban la existencia de sitios Web con contenidos xenofóbicos o nazis, si en los Estados Unidos esto es perfectamente legal, o que en nuestro país se abogue por la no comercialización del sexo si este negocio arroja miles de millones de ganancias en otros países,³ y por tanto tienen los recursos y la justificación de la libertad de expresión para vetar cualquier ley que intente limitar este denigrante mercado. Otro de los ejemplos son los relativos a la pornografía infantil, en algunos países se prohíbe la distribución y la recepción, en otros solo la distribución y en algunos ninguna de ambas cosas.

En varios países se han tomado iniciativas de ley que han terminado en un rotundo fracaso, ejemplo de esto es la controvertida Acta de Decencia de las Comunicaciones de EE.UU., promovida en 1995 por los senadores James Exon y Slade Gorton en el marco de la nueva ley de telecomunicaciones de ese país⁴. En Alemania existe una ley sobre los servicios electrónicos que establece que los ISP (Proveedores de acceso a Internet) pueden ser perseguidos por dar la oportunidad a sus usuarios de visitar recursos con contenidos ilegales⁵, esta ley por ejemplo no es del todo efectiva ya que muchas veces los proveedores de acceso a Internet no están en condiciones de saber cuales contenidos están visitando sus usuarios⁶. Ejemplos un poco más efectivos son la experiencia del Reino Unido: en este país los ISP han adoptado mecanismos de regulación a través de códigos de conducta y la creación de un organismo dedicado a recibir denuncias sobre contenidos ilícitos llamado, Safety Net Foundation⁷.

Esta preocupante situación internacional ha obligado a la sociedad a buscar posibles vías de solución a esta problemática. Hasta el momento son las soluciones técnicas las únicas que han logrado cierto nivel de éxito a la hora de regular un tanto qué contenidos son accesibles por los usuarios. Una de las soluciones a esta problemática es el uso de sistemas de filtrado, los cuales pueden ser basados en protocolos o autónomos. Los sistemas basados en protocolos, en general, hacen uso de la Plataforma para la Selección de Contenido en Internet (PICS), desarrollada por el consorcio W3C como un medio para el intercambio de información sobre clasificación. Los sistemas autónomos determinan qué sitios van a ser bloqueados según criterios establecidos

³ Estadísticas publicadas por la: "Fundación Misioneros del Sagrado Corazón en el Perú".

⁴ "Telecommunications Act of 1996", Pub. LA. No. 104-104, 110 Stat. 56 (1996)

⁵ "Ley de Telecomunicaciones de Alemania", en el boletín oficial federal I N° 29.

⁶ Rodríguez Pereda, Katitza: "Tratamiento jurídico de los contenidos nocivos e ilícitos que circulan en Internet"

⁷ Villate, Javier, 2001, "Libertad de expresión en Internet".



por el fabricante; generalmente combinan el uso de listas de inclusión y exclusión con la búsqueda de palabras claves.

En esta ponencia, se describe un Sistema de Filtrado Adaptativo que su principio de funcionamiento se basa en el uso de listas blancas y negras, previendo además la inclusión de un subsistema de Recuperación y Clasificación Automática de Información. Este último sería el que le daría la posibilidad de adaptarse a la naturaleza dinámica que posee Internet.

En el trabajo se exponen algunas estadísticas que demuestran la necesidad real de utilizar estos sistemas, continuando con una serie de conceptos básicos, sin los cuales la comprensión de lo que sigue se tornaría difícil para el lector no entendido en el tema. De esta forma se describe el principio de funcionamiento de este Filtro Adaptativo, los componentes que posee, el mecanismo de recuperación-clasificación y algunos de los resultados parciales obtenidos.

¿Por qué utilizar Sistemas de Filtrado?

Los sistemas de filtrado surgen como una alternativa para proteger a los usuarios de Internet de los contenidos inadecuados e ilícitos que hoy pululan en esta red. A continuación mostramos algunas estadísticas que avalan lo anterior 13:

- La pornografía en Internet genera aproximadamente mil millones de dólares anualmente, con proyecciones de crecimiento en ascenso, de \$5-7 mil millones en los próximos 5 años.
- Existen casi un millón de sitios que fomentan el uso de drogas.
- Existen 4.2 millones de sitios pornográficos; 12 % del total de sitios Web.
- Existen 372 millones de páginas pornográficas.
- Se realizan 68 millones de peticiones diarias a los motores de búsquedas que están relacionadas con la pornografía; el 25% de las búsquedas.
- Se envían 2.5 billones diarios de correos electrónicos con contenidos pornográficos, 8% del total.
- Se realizan 1.5 billones de descargas (P2P) mensuales de materiales pornográficos (35% de las descargas).
- 72 millones de usuarios visitan sitios Web con contenidos pornográficos al año.
- El 20% de los hombres y el 13% de las mujeres admiten acceder a sitios pornográficos en el centro donde laboran.
- El 72% de los hombres y el 28% de las mujeres que visitan sitios pornográficos adquieren anomalías en la personalidad.
- Se estiman en más de 100 mil los sitios Web que ofrecen pornografía infantil.
- El 90% de los niños que han visto escenas sexuales fuertes lo hicieron de forma accidental al buscar juguetes o marcas de ropa en la Web.
- Más de 190 mil sitios Web utilizan nombres de marcas populares para engañar a los menores de edad.
- Según estudios realizados en Europa, el 44% de los menores que navegan con regularidad se han sentido acosados sexualmente en varias ocasiones.

Conceptos Básicos

Filtro de contenido, uno o más elementos de software que operan juntos para evitar que el usuario acceda a contenidos disponibles en Internet. En el proceso intervienen dos componentes. Clasificación: se utilizan juicios de valor para categorizar recursos sobre la base de su contenido. Estas clasificaciones pueden utilizar simples distinciones "permitido/no permitido", como las que se encuentran en programas como CyberSitter o NetNanny, o pueden tener muchos valores, como en el caso de los sistemas de clasificación basados en PICS (Plataforma para la Selección del Contenido en Internet).

Filtrado: cada solicitud de información, es analizada por el software de filtrado. Si el recurso está en la lista de los "no permitidos" o no tiene la clasificación PICS adecuada, el software de filtrado le informa al usuario que el acceso le ha sido denegado y el navegador no muestra el contenido de la Web original.



Filtrado autónomo, los primeros filtros de contenido eran sistemas autónomos rígidos, que consistían en mecanismos que determinaban los sitios que debían bloquearse, junto con el software que realizaba el filtrado, todo ello suministrado por un único vendedor. Aunque hoy en día han evolucionado y son ampliamente utilizados.

Filtrado basado en protocolos, estos sistemas consisten en elementos de software que utilizan estándares establecidos para comunicar la información de clasificación a través de Internet, en la mayoría de los casos en forma de etiquetas que se insertan dentro del código de las páginas Web. A diferencia de los sistemas autónomos, los sistemas basados en protocolos no contienen ninguna información sobre los sitios que serán bloqueados. Los sistemas basados en protocolos sólo saben cómo encontrar esta información en Internet y cómo interpretarla, en estos son los usuarios quienes deciden lo que será bloqueado.

Categorías, lo que agrupa un conjunto de recursos atendiendo a un contenido común.

Listas blancas, tienen sentido en los filtros del tipo autónomo, es una lista de inclusión. Con esta lista se persigue el objetivo de permitir el acceso a los recursos que estén contenidos en ella. En algunos casos solo se permite el acceso a estas listas, las cuales están constituidas por un conjunto de sitios tales que garanticen una navegación segura. Tienen aplicación en lugares donde los usuarios sean fundamentalmente niños.

Listas negras, listas que surgen con los filtros autónomos, es una lista de exclusión. Con esta lista se persigue el objetivo de negar el acceso a los recursos que estén contenidos en ella. En algunos sistemas de filtrado sólo se niega el acceso a estas listas, las cuales están constituidas por un conjunto de sitios de contenidos dudosos. El campo de aplicación es más amplio, se usan en todo lugar donde se requiera impedir el acceso a algunos recursos.

Listas grises, es un tipo de lista en la que los recursos tienen asociado un determinado nivel dentro de una misma categoría, la cantidad de niveles depende del filtro en cuestión. Los puntos extremos de esta lista vienen siendo el equivalente a las listas negras y blancas. En general este tipo de lista le da la posibilidad al sistema de filtrado de disponer de un rango de decisión más amplio, a expensas de una mayor complejidad en los algoritmos que utiliza para construir la lista.

“Pornografía, es la exhibición de genitales y actos sexuales de toda índole, donde abundan imágenes sadomasoquistas, la pedofilia y otras aberraciones, individuales o en grupo. Es una industria millonaria que vende sexo cosificando y envileciendo al hombre y a la mujer. Es explotación de los seres humanos y sobre todo de la mujer”⁸.

“Pornografía infantil, toda representación, por cualquier medio de comunicación, de un niño o niña menor de 18 años de edad, o con aspecto de niño o niña, involucrado en actividades sexuales reales o simuladas, de manera explícita o sugerida, con cualquier fin. El consentimiento por parte de los niños y las niñas para participar en pornografía no es válido de acuerdo con lo establecido en la normativa internacional.”⁹

Contenidos ilícitos, "entendemos por contenidos ilícitos aquellos susceptibles de entrar bajo el ámbito de aplicación de las normas penales y/o civiles de un ordenamiento jurídico determinado" ¹⁰. Como vemos, la calificación de ilícito dependerá de la legislación vigente en cada país. Como ejemplo típico de este contenido se toma a la pornografía infantil.

Contenidos nocivos, "cuando hablamos de contenidos nocivos nos estamos refiriendo a aquellos materiales que sin ser ilícitos pueden resultar perjudiciales u ofensivos para ciertos grupos o categorías de personas. Puede referirse a contenidos sobre creencias políticas, religiosas, culturales, sexuales, etc., que

⁸ Lazo Urbina, E.; Marín González, H. y Marroquín Molina, S. (1997). "La pornografía"

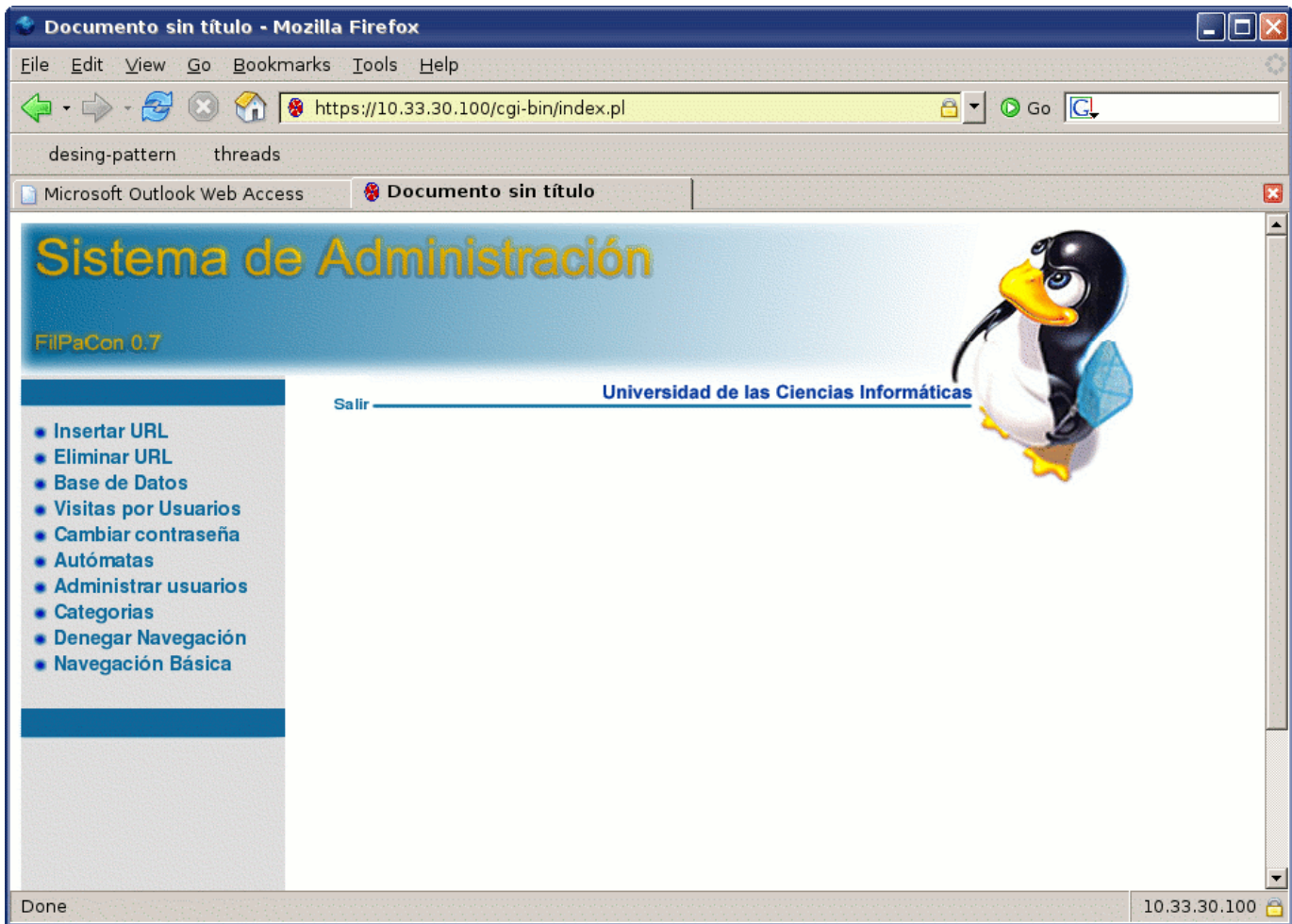
⁹ Palacio B., Diego; Pinto D., Martha E. Londoño S., Beatriz: "Criterios de clasificación de páginas en Internet con contenidos de pornografía infantil" Ed. Instituto Colombiano de Bienestar Familiar, ISBN 9 58-623-066-X, 2004.

¹⁰ Álvarez Valenzuela, Daniel, 2004, "Libertad de Expresión en Internet y el control de Contenidos Ilícitos y Nocivos" Universidad de Chile, Facultad de Derecho, Centro de Estudios en Derecho Informático.



eventualmente pueden ser considerados ofensivos respecto de terceros" 15. Al definirse de esta forma un contenido nocivo está sujeto a diversos factores culturales. "El ejemplo más universal de este tipo de contenidos corresponde a la pornografía, ampliamente difundida a través en Internet" 15.

Contenidos adecuados, aquellos que no entran en ninguna de las definiciones antes mencionadas. Aquellos que son perfectamente legales y que no afectan la moral de ninguna persona sin importar edad, sexo, raza, cultura, etc.



Principio de Funcionamiento

El sistema parte del principio de disponer de una Base de Datos (BD) en la cual se encuentran un sinnúmero de URLs preclasificadas en distintas categorías atendiendo al contenido que albergan. Las categorías a su vez son clasificadas por el administrador del sistema según el tipo de contenidos al que se estima pertenecen y las características propias del lugar (cultura, normas, regulaciones, leyes vigentes, etc). Así una categoría puede pertenecer a uno de los siguientes tipos: ilícita, nociva o adecuada y a su vez las categorías pudieran ser: pornografía, terrorismo, violencia, juegos, noticias, deportes, salud, otras más y todas las que el administrador estime conveniente adicionar.

Todo comienza cuando un cliente (ver **Figura 2**) le solicita un recurso de Internet al servidor Squid-Proxy, el cual es parte inseparable del sistema, a partir de este momento el Squid le entrega al Subsistema de Filtrado



los parámetros de la solicitud: identificador del usuario, dirección IP de origen y URL, comienza así un proceso de interacción con la BD el cual dependiendo de: características usuario, edad y política de navegación, la existencia o no de la URL en la Base de Datos, las categorías y tipo de contenido al que pertenece la URL, se toma una decisión (ver **Tabla 1**) sobre si el recurso solicitado se le entrega al usuario o si en su lugar se le entrega una página (ver **Figura 1**) en la que se le explican los motivos por los cuales se le negó el acceso a dicho recurso. En el punto actual de desarrollo aun no se ha incorporado el Subsistema de Recuperación y Clasificación de Información (RCI, ver **Figura 3**), se prevé que este sistema funcione de manera independiente pero tomando como uno de los posibles puntos de partida la información que le brinde el Subsistema de Filtrado.

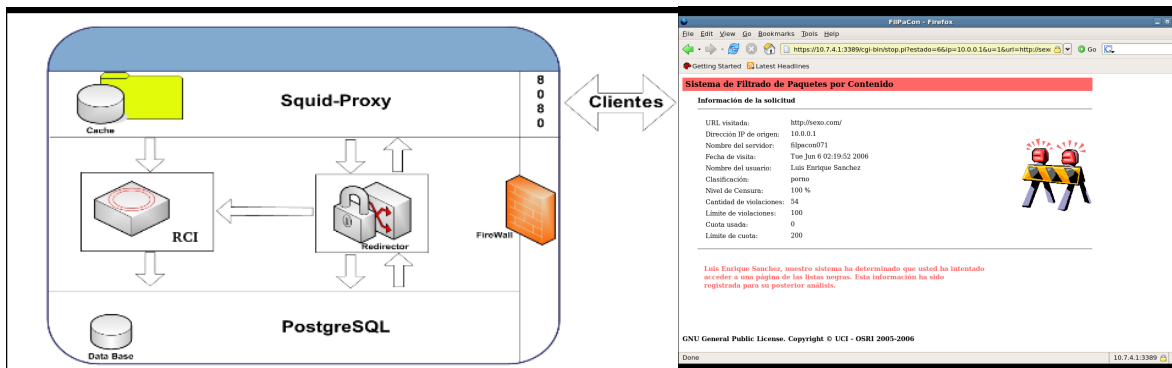


Figura 2: Esquema funcional del sistema

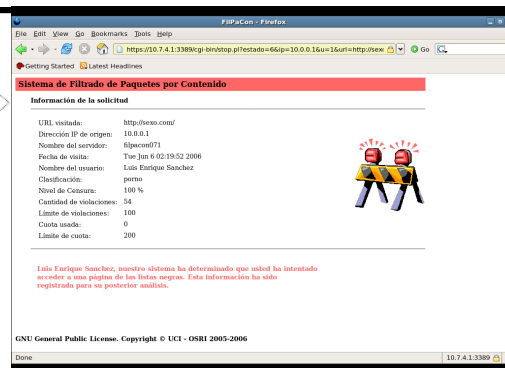


Figura 1: Página de negación de solicitud

Todo se ha diseñado para que pueda funcionar de manera centralizada o distribuida; tal como se muestra en los esquemas de las **Figuras 3 y 4** respectivamente. En la primera variante: el Squid-Proxy, la Base de Datos, el Subsistema de Filtrado y el Subsistema de Recuperación y Clasificación de Información estarán funcionando en el mismo servidor. En la segunda variante: Squid y el Subsistema de Filtrado se encuentran en un primer servidor, la Base de Datos en un segundo servidor y el Subsistema de Recuperación y Clasificación de Información en un tercer servidor independiente. Esta variante es para lugares donde se requiera alto desempeño.

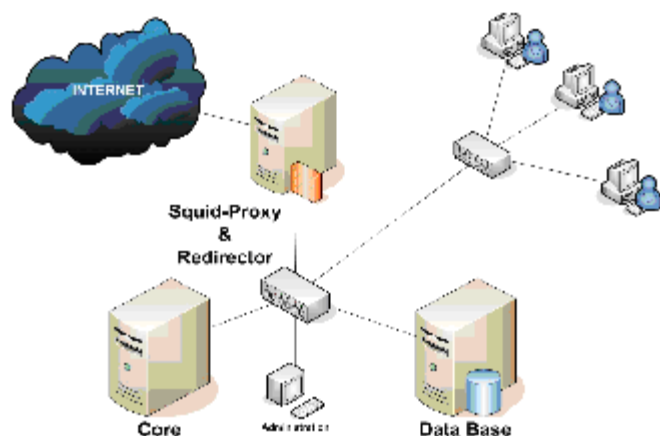


Figura 4: Esquema centralizado del sistema

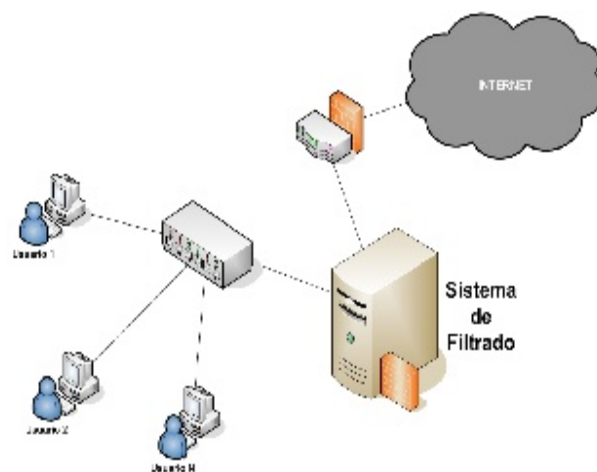


Figura 3: Esquema distribuido del sistema

Detalles de los componentes del sistema

El sistema esta compuesto por:

- Subsistema de instalación
- Subsistema de filtrado
- Subsistema de administración
- Base de Datos
- Subsistema de recuperación y clasificación de la información

A continuación se explicarán algunas características de los componentes del sistema:

Subsistema de Instalación

El objetivo principal de este Subsistema es lograr que el proceso de instalación tenga como características las siguientes:

- Instalación desatendida e intuitiva.
- Detección automática y soporte amplio de hardware.
- Instalación y configuración automática del sistema.
- Exigencias de conocimiento, mínimas para el administrador.

Subsistema de Filtrado

El objetivo principal de este Subsistema es tomar una decisión sobre que contenido será bloqueado y cual no, siempre teniendo en cuenta la información que le facilita la Base de Datos. La **tabla 1** expresa la relación que existe entre las características de usuario: edad y política de navegación, y el tipo de contenido al que pertenece la categoría de la URL que se está procesando. Con este esquema se logra una gran flexibilidad del sistema, permitiendo que pueda adaptarse a disímiles situaciones.

Subsistema de administración

El objetivo de este subsistema es brindar a los administradores una GUI intuitiva que permita realizar las tareas de configuración y ajustes necesarios para el correcto funcionamiento y adecuación a las características propias del lugar. Esta interfaz no necesita de la instalación de ningún software adicional para su explotación, es una aplicación Web que funciona con cualquier navegador (Internet Explorer, NetScape, Mozilla, etc) disponible.

Esta interfaz le brinda al administrador un conjunto de funcionalidades imprescindibles para garantizar: gestión de los usuarios, gestión de las categorías existentes en la BD, gestión de las URL's, la configuración del sistema y la generación de reportes sobre el uso de Internet.



Edad	Política de Navegación	Decisión a tomar	Consulta en la Base de Datos
Menor / Mayor	Ninguna	No tiene acceso a Internet	No se hace nada
Menor / Mayor	Básica	Sólo tiene acceso a las direcciones definidas como básicas	La tabla de dominios básicos define las direcciones que pertenecen a esta política de navegación
Menor	Rest restrictiva	Permite solamente las categorías adecuadas	Si la dirección no existe en la base de datos se deniega por defecto
Mayor	Rest restrictiva	Permite solamente las categorías adecuadas y nocivas	Si la dirección no existe en la base de datos se deniega por defecto
Menor	Permisiva	Deniega solamente las categorías ilícitas y nocivas	Si la dirección no existe en la base de datos se deniega por defecto
Mayor	Permisiva	Deniega solamente las categorías ilícitas	Si la dirección no existe en la base de datos se deniega por defecto
Menor / Mayor	Plena	Acceso Total	No importa

Tabla 1. Política de navegación del sistema

Módulos y Funcionalidades que brinda la interfaz de administración.

Módulos:

- Gestión de Usuarios
- Gestión de categorías
- Gestión de URL's
- Configuración del sistema.
- Reportes

Funcionalidades:

- Insertar usuario.
- Eliminar usuario.
- Actualizar los datos del usuario.
- Cambiar contraseña.
- Denegar la navegación.
- Habilitar la navegación.
- Insertar nueva categoría.
- Eliminar categoría existente.
- Seleccionar tipo de contenido para cada categoría.
- Insertar categoría a una URL.
- Actualizar categoría de una URL.
- Comprobar categorías a la que pertenece una URL.
- Eliminar una categoría de una URL.
- Eliminar URL.
- Configurar sección del sistema.
- Configurar sección de la Base de Datos.



- Configurar sección de la Interfaz de Administración.
- Restablecer configuración.
- Establecer los dominios de navegación básica.
- Incidencias por usuarios.
- Ranking de las URL's denegadas.
- Ranking de usuarios por incidencias.
- Ranking de días de la semana por incidencias (escoger rango de fecha).
- Cantidad de incidencias por tipo de usuarios (escoger rango de fecha).
- Reporte de la cantidad de sitios por categoría.
- Reporte de posibles regulaciones incorrectas.

Base de datos

El objetivo que se persigue con este medio es almacenar las listas de URL's y la información relacionada con los usuarios y sus actividades. En la **tabla 2** se muestran las categorías, una descripción y la cantidad de URL que están clasificadas hasta la fecha.

Categoría	Descripción	Cantidad
Pornografía	Sitios con contenidos pornográficos.	830999
Regiones	Sitios relacionados con algunas regiones del planeta.	670954
Mundo	Sitios de contenidos variados relacionados con países.	503416
Artes	Sitios relacionados con las artes.	289227
Sociedad	Sitios relacionados con diferentes temas de la sociedad (religión, política, organizaciones no gubernamentales, etc).	262964
Negocios	Sitios para negocios.	251491
Noticias	Agencias de noticias on-line.	229342
Computadoras	Sitios de negocios de computadoras.	138149
Recreación	Sitios de contenido recreativo.	118605
Tiendas	Tiendas virtuales.	116699
Deportes	Sitios relacionados con el deporte.	107539
Ciencias	Sitios relacionados con las ciencias.	103590
Referencia	Sitios de referencia bibliográfica.	64745
Salud	Sitios que promueven la salud.	64415
Juegos	Sitios dedicados a dar publicidad a los juegos.	61772
Adolescentes	Sitios para niños y adolescentes.	38997
Hogar	Sitios relacionados con artículos para el hogar.	33227
Spyware	Sitios que capturen información de los usuarios.	10453
Drogas	Sitios donde se promueve el uso de drogas.	2029
Lotería	Sitios que promueven los juegos de azar.	1223
Publicidad	Sitios publicitarios.	591
Hackeo	Sitios que publican herramientas o contenidos para ataques informáticos.	373
Audio_video	Sitios relacionados con Audio o video.	359
Redirector	Sitios que burlan los sistemas de filtrado.	329
Violencia	Sitios con contenido agresivo.	319
Warez	Sitios relacionados a la piratería informática.	87
Webmail	Sitios que brindan servicios de correo on-line.	47
Foros	Foros de discusión en Internet.	40
Radio	Emisoras de radio en Internet.	25
Materiales_peligrosos	Fabricación de explosivos, venenos, etc.	18
Educación_sexual	Sitios de educación sexual.	33
Total	Total de categorías clasificadas	902027

Tabla 2. Categorías con que cuenta el sistema



Subsistema de Recuperación y Clasificación de la Información

Dado el dinamismo de Internet las listas blancas y negras constantemente estarán desactualizadas. Surge así la necesidad de contar con un mecanismo que permita realizar el análisis de los contenidos presentes en las páginas Web para apoyar la toma de decisiones y hacer frente al continuo crecimiento y expansión de la red de redes. Para ello se está desarrollando un Subsistema que será el encargado de recuperar y clasificar la información. Este Subsistema está compuesto por un módulo de recuperación de información, un clasificador de textos, un clasificador de imágenes y un módulo decisor; este último entrega un resultado de categorización combinando varios factores, entre ellos el resultado de ambos clasificadores. Aunque actualmente no está integrado al sistema y se encuentra en franco proceso de desarrollo, se le han realizado pruebas y se continúa perfeccionando.

Módulo de Recuperación de Información

Para poder llevar a cabo las acciones descritas anteriormente es necesario contar con un mecanismo que permita recuperar el contenido de las páginas Web. Para esto se desarrolló un robot (spider) que navega de manera automática por la Web a partir de una serie de sitios que se le especifican en un fichero o que simplemente recupera la información de una URL en particular para la cual sea invocado.

Módulo de Clasificación Automática de Texto

Las numerosas desventajas de la búsqueda de palabras claves conduce irremediablemente a la necesidad de encontrar un algoritmo más eficiente para clasificar los contenidos semánticos de las páginas Web. Para ello se empleó la categorización automática de textos lo que puede contemplarse como un proceso de aprendizaje, durante el cual un programa capta las características que distinguen una clase o categoría de las demás. Antes de comenzar la clasificación se verifica que el idioma esté soportado por el clasificador. Si el idioma identificado es válido se realiza un proceso de Indexación automática que persigue lograr una representación del documento que pueda ser entendible por el ordenador.

Esta representación se hace en forma de vectores de pesos de términos que son usados en la clasificación, la cual se materializa hallando la medida de similitud entre estos vectores y los vectores representativos de cada categoría. Esto produce como salida la probabilidad que tiene el documento en cuestión de pertenecer a cada una de las categorías, siendo clasificado en la que mayor probabilidad tenga. Actualmente están soportados los idiomas español e inglés.

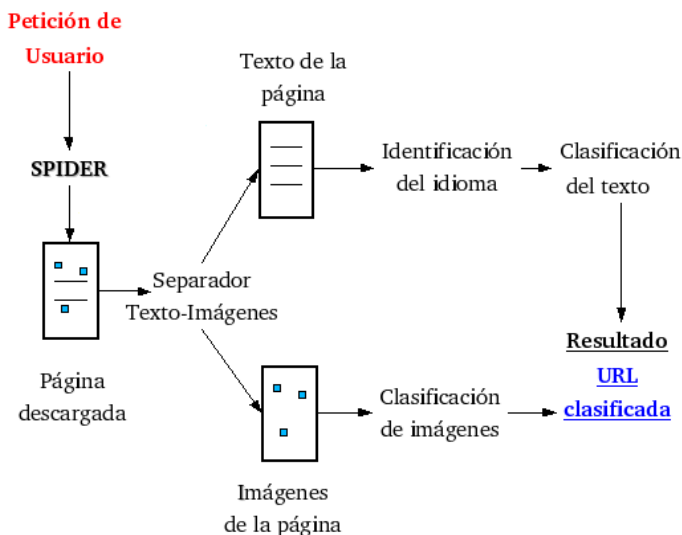


Figura 5: Subsistema de Recuperación y Clasificación de la Información



Módulo de clasificación Automática de Imágenes Pornográficas

Aunque el análisis del texto que está contenido en las páginas Web con técnicas de inteligencia artificial es una solución que elimina los problemas asociados a la búsqueda de palabras claves aún persiste uno de ellos y es la imposibilidad de clasificar la información contenida en las imágenes presentes en el hipertexto.

El objetivo de este módulo es el de apoyar la toma de decisiones mediante el análisis de las imágenes contenidas en las páginas Web para determinar si son pornográficas o no. Este módulo está desarrollado siguiendo la estructura básica de las aplicaciones de reconocimientos de patrones: adquisición, segmentación, extracción de características y entrenamiento-clasificación. En este proceso se toma el color de la piel como característica primaria. Hasta ahora se ha utilizado el método de los n-vecinos más cercanos de conjunto con redes neuronales; obteniéndose en las pruebas realizadas una efectividad de un 60%.

Resultados

Este sistema aunque se encuentra en desarrollo posee una serie de funcionalidades básicas que han permitido obtener los siguientes resultados experimentales:

- Se ha evitado el acceso a sitios con contenidos inadecuados pre-clasificados.
- El tiempo que tarda hacer la consulta en la base de datos es despreciable, en el orden de los 20 milisegundos.
- Algunas URL's han sido denegadas incorrectamente debido a que el dominio al que pertenecen alberga algunas páginas con contenidos inadecuados y por tanto se ha bloqueado el dominio completo; en algunos casos se ha cambiado la clasificación del dominio para permitir la navegación por este.

El 10 de febrero del 2006 se realizó una prueba en los laboratorios docentes de la Facultad 10 de la Universidad de las Ciencias Informáticas, el objetivo fundamental de la misma era comprobar el funcionamiento adecuado bajo condiciones de carga media en el servidor experimental donde está instalado el sistema; esta se realizó con 120 estudiantes distribuidos en 6 laboratorios docentes con 20 PC. Los resultados de esta prueba se recogen en la Tabla 3. Las características del servidor:

CPU: Intel(R) Pentium(R) 4 CPU 2.40GHz, 8KB L1
Cache, 512 KB L2 Cache.
MM : 1GB Memoria Principal
Red : Ethernet interface 100MB/s
HDD: 74 GB

Proceso	CPU %	Memoria %
Squid Proxy	15.703	5.7752
Subsistema de filtrado	10.080	1.5670
Gestor Base de Datos	3.0562	0.5014
Servidor Web	1.0066	0.3499

Tabla 3. Resultados de la prueba de carga

Al finalizar la prueba se realizó una encuesta a los 120 estudiantes; donde se anularon 3 por respuestas incongruentes. Los resultados fueron los siguientes:

- 89% respondió que tenía experiencia anterior de navegación por Internet, el 11% que nunca lo había hecho.
- 23% respondió que se le denegó alguna página por error, 71% que no, 6% Sin respuesta.
- Velocidad de navegación: muy rápida 13%, rápida 59%, normal 25%, lenta 2%, muy lenta 1%. Comparado con el servidor de internet de la Universidad.
- El promedio de uso del procesador fue de un 30%.



Requerimientos del sistema

El sistema está diseñado para funcionar sobre la distribución Debian GNU/Linux. Los requerimientos se muestran en la Tabla 4.

Cantidad de usuarios	CPU	Memoria	HDD
0 – 50	450 MHz	64 MB	2GB
50 – 100	1.7 GHz	256 MB	20 GB
+ 100	3.2 GHz	1 GB	80 GB

Tabla 4. Requerimientos del sistema



Conclusiones

A pesar de que el sistema se encuentra aun en fase de desarrollo si permite palpar resultados parciales en las pruebas que se le han hecho. Los tiempos de respuesta, de la Base de Datos aun teniendo millones de campos son mínimos, sin embargo aun faltan por hacer pruebas más profundas en este sentido. El objetivo fundamental de describir el Sistema de Filtrado Adaptativo que se desarrolla en la Universidad de las Ciencias Informáticas, al mismo tiempo que se brinda una panorámica sobre la problemática que existe a nivel mundial sobre la regulación de los contenidos ilícitos y nocivos ha sido cumplido con la redacción de este trabajo. Sin embargo aun queda por recorrer el camino que convertiría a esta solución en una realidad palpable en toda su dimensión: los algoritmos de IA que garanticen una Base de Datos lo suficientemente actualizada para garantizar una navegación confiable para los usuarios.



Riesgos de seguridad utilizando system() o popen()

Jorge Luis Betancourt González
jlbetancourt@estudiantes.uci.cu
Universidad de las Ciencias Informáticas

Uno de los agujeros de seguridad que todo programador debe tener en mente incluye la utilización de la shell para ejecutar otros programas. Como ejemplo, vamos a considerar un servidor diccionario. Este programa está diseñado para aceptar conexiones vía Internet: cada cliente envía una palabra, y el servidor dice si es o no un término válido del idioma Inglés. Debido a que cada sistema GNU/Linux viene con una lista de aproximadamente 45 000 palabras en inglés en /usr/share/dict/words, una forma inteligente de resolver este problema es invocando al programa grep. Por ejemplo:

```
% grep -x word /usr/share/dict/words
```

En este caso, word es la palabra que el usuario está buscando. El código de salida de grep dirá si esa palabra aparece o no en /usr/dict/words.

```
/*
 * grep_dictionary.c
 */
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
/*
 * Returns a nonzero value if and only if WORD appears in /usr/dict/words
 */
int grep_for_word(const char* word)
{
    size_t length;
    char *buffer;
    int exit_code;
    /*
     * Build up the string 'grep x WORD
     * /usr/dict/words'.
     * Allocate the string dynamically to avoid buffer overruns.
     */
    length = strlen("grep x") + strlen(word) + strlen("/usr/share/dict/words") + 1;
    buffer = (char*)malloc(length);
    sprintf(buffer, "grep x %s /usr/dict/words", word);
    /*
     * Run the command, using the system() function.
     */
    exit_code = system(buffer);
    /*
     * Free the buffer
     */
    free(buffer);
    /*
     * If grep returned 0, then the word was present in the dictionary, otherwise not.
     */
    return (exit_code == 0);
}
```



Nótese que calculando el número de caracteres que necesitamos y entonces reservando el buffer dinámicamente, nos aseguramos que nunca exista una sobrecarga del buffer. Desafortunadamente el uso de la función `system()` es inseguro. Esta función invoca al shell estándar del sistema para ejecutar el comando y entonces devolver el valor del código salida. ¿Pero qué sucedería si un usuario malicioso enviara una “palabra” como la que sigue a continuación?:

```
% foo /dev/null; rm -rf /
```

En este caso el servidor ejecutaría el siguiente comando:

```
%grep -x foo /dev/null; rm -rf / /usr/dict/words
```

Ahora el problema es obvio. El usuario ha transformado un comando, la invocación a `grep`, en dos comandos porque la shell trata al punto y coma (;) como un separador de comandos. El primer comando sigue siendo una inofensiva invocación a `grep`, pero el segundo eliminará todos los ficheros del sistema completo. Incluso si la aplicación no está ejecutándose como root, todos los ficheros que puedan eliminarse por el usuario que está ejecutando el servidor serán eliminados.

El mismo problema puede presentarse con la función `popen()`, la cual crea una tubería entre el proceso padre y el proceso hijo, pero sigue utilizando la shell para ejecutar el comando. Existen dos formas de prevenir este problema: uno es utilizar la familia de las funciones `exec()` en vez de `system()` y `popen()`. Esta solución evita los problemas porque los caracteres que el shell trata especialmente (como el punto y coma en el ejemplo anterior) no son tratados especialmente cuando aparecen en la lista de argumentos de una llamada a `exec()`. Por supuesto, usted puede ver las ventajas sobre `system()` y `popen()`. La otra alternativa es validar la cadena y asegurarse que es benigna.

En el ejemplo del servidor diccionario, usted debe asegurarse de que la palabra proveída contiene solamente caracteres alfabéticos, utilizando la función `isalpha()`. Si no contiene otro carácter entonces no hay forma de engañar a la shell para ejecutar un segundo comando. No se recomienda implementar el chequeo de forma tal que busque los caracteres peligrosos o inesperados, siempre es más seguro chequear de forma explícita los caracteres que usted conoce que son seguros que tratar de anticipar todos los caracteres que pueden causar problemas.



¿Qué proyectos usan y desarrollan Software Libre en la UCI?

Ing. Abel Meneses Abad
abelma@uci.cu
Proyecto UNICORNIOS

Con frecuencia se habla ya del software libre en la Universidad de las Ciencias Informáticas (UCI), y muchos se preguntan: ¿qué se está haciendo?, ¿qué proyectos existen? Para los jóvenes investigadores este artículo intentará mostrar las escasas, desorganizadas, deficientes y en ocasiones incluso inexistentes fuentes de información a las que deben acudir para experimentar la migración a SWL.

¿Dónde saber en qué proyectos, facultades, etc, se está usando SWL?

Wikiprod¹¹. El primero de los lugares a donde nos podemos dirigir para conseguir información sobre los proyectos es la wiki de la UCI; allí se encuentra, en dependencia de la cultura de los líderes de los proyectos, gran parte de la información. Ejemplo: en la portada inicial podemos encontrar una categoría llamada Producción UCI. Existe en esta categoría un enlace al "portal de producción", pero desafortunadamente la información disponible es "0", los enlaces están abandonados.

Al realizar una investigación tratamos de acceder a todos los puntos importantes que nos lleven a cumplir el objetivo final; si intentamos encontrar las fuentes de información es por ello importante hacer uso de cada link. Así el ejemplo descrito anteriormente nos llevó a la dirección: <http://dirproduccion2.uci.cu/>; deduciendo, probamos también los enlaces: <http://dirproduccion1.uci.cu/> (NO EXISTE); <http://dirproduccion3.uci.cu/> (Sí existe); también se intentó con los números 4 y 5, pero no existen. Volveremos entonces a esta fuente de información. Regresemos a la información disponible en la wikiprod. Allí encontrará usted los eventos, el grupo de trabajo de Producción de la FEU-UJC, la categoría Producción UCI que nos lleva a más información aún.

¿A quiénes preguntar por los proyectos libres?

El autor considera que la disponibilidad en la Web de información a veces no confiable hace necesario, para la credibilidad de los resultados de esta investigación, poner a disposición de todos los nombres de las personas a las cuales se les envió el siguiente correo:

"En estos momentos realizo una investigación sobre qué proyectos pueden estar trabajando en SWL. ¿Conoce usted los lugares en la Web UCI, líderes de proyecto, o los nombres de proyectos con los cuales conformar mi investigación?"

*David Leyva Leyva, Vicedecano Producción Ftad X
Loan Joa Matos, Vicedecano Producción Ftad 4
Pedro Y. Piñeiro Pérez, Vicedecano Producción Ftad 3
Aurelio Antelo Collado, Vicedecano Producción Ftad 6
Grupo de Producción FEU – UCI.*

A pesar de la simplicidad de la información a enviar que se respondía en el caso más organizado, con un link hasta un lugar donde los investigadores puedan continuar el proceso de búsqueda, solo dos compañeros de la FEU y un vicedecano de producción respondieron. Lo que indica la primera dificultad de los proyectos libres, poca promoción y divulgación.

Los compañeros escogidos en esta lista no están de manera casual, las cuatro facultades mencionadas son las que mayor desempeño han tenido hasta el momento, a consideración del autor, en desarrollos sobre plataformas libres (Eclipse, Zope/Plone, PERL, etc.)

¹¹ <http://wiki.prod.uci.cu>



¿Qué proyectos de SWL existen hoy en la UCI?

Para decir si efectivamente cada proyecto de los que encontramos en la Web de la UCI existe aún, dado que las actualizaciones no son muy constantes, es necesario enviar correos a sus líderes y cuestionar el seguimiento al proyecto. Ponemos entonces a disposición de la comunidad universitaria la información que motivara esta investigación.

Nota: Algunos proyectos más pueden ser encontrados en la página de la Facultad 3¹², sobre todo relacionado con desarrollo para sistemas de gestión basados en tecnologías Zope/Plone.

Nombre del Proyecto	Descripción	e-mail del líder del proyecto	Tecnologías
Nova GNU/Linux	Desarrollo de una distribución libre para la República de Cuba.	agoni@estudiantes.uci.cu	Gentoo, C++, GTK
FILPACON	Desarrollo de un filtro de internet para Cuba, y empresas.	alaing@uci.cu	Debian, Perl, PHP, MySQL
RINDE	Red de Integración y Desarrollo, una red para el trabajo, la promoción del SWL en Venezuela.	jpernia@uci.cu	GForge, wiki
Biblioteca Nacional	Desarrollo de un sistema de gestión documental.	lgongora@estudiantes.uci.cu	Zope/Plone
Portales	Desarrollo de portales Web para la intranet nacional.	maikelm@uci.cu leansy@uci.cu	Drupal/PHP & Zope/Plone/Python
UNICORNIOS	Soporte, y desarrollo de aplicaciones, para la migración a SWL.	abelma@uci.cu yvillazon@estudiantes.uci.cu	C++, PHP, Python, Perl, Bash
Scada	Desarrollo en JAVA de un simulador para PDVSA	rene@uci.cu	Java
Intranet PDVSA	Desarrollo de una intranet Corporativa	sergioenrique@uci.cu	PHP
Software de Gestión para la Oficina Nacional de Recursos Mineros	Software para controlar lo referido a yacimientos minerales y su explotación en el país.	vito@uci.cu	J2EE(JBOSS), Eclipse
Automatización de las refinerías de PDVSA	Simulación de una refinería. Automatización y control. Software de Gestión.	lperezm@uci.cu	PostgreSQL, PHP, Python+GTK
Aduana General de la República	Proyecto integral de desarrollo y estrategia de migración de la aduana general de la república.		

Tabla 5. Descripción de los proyectos que usan y desarrollan SWL en la UCI

¹² <http://facultad3.uci.cu>



Conclusiones

El auge de los proyectos que desarrollan en software libre es cada día mayor. Podemos encontrar hoy muchos desarrollos basados en PHP, Java y Python. No hemos de dudar de que, en breve tiempo, estos se harán sobre sistemas operativos libres. Dejamos a los lectores la posibilidad de solicitar una nueva revisión de esta investigación, que en pocos meses podría arrojar nuevos proyectos libres dentro de la UCI.



Tocororo GNU/Linux

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

Tocororo GNU/Linux es una distribución que se desarrolla en ambiente comunitario contando con el apoyo de las Comunidades de Desarrollo de la UCI. Está destinada a la educación y tiene como objetivo implantarse en la enseñanza primaria y secundaria, así como en los Joven Club de Computación y Electrónica; está pensada para PC's con pocos recursos de hardware, destinada al usuario final y con entorno gráfico Xfce4.



Figura 7: Propuesta inicial de logo



Figura 6: Otra propuesta

El logo en sí (segunda propuesta) es la imagen que aparece encima de las letras. A continuación, una síntesis de lo que estas expresan:

- La forma de la rama que sostiene al tocororo intuye la forma de nuestra isla, y las hojas de conjunto con ésta resumen la frase: "Cubano por naturaleza...", que propusiera inicialmente Tuxifer (Dairo Reyes), uno de nuestros coordinadores.
- El Tocororo expresa el nombre de la distro
- La filosofía del Software Libre puede deducirse de la "personalidad" del tocororo, pues para muchos es sabido que esta ave endémica de nuestro país no puede vivir en cautiverio.
- El sello de identidad final se cierra con el empleo de los colores, manejados con la intencionalidad marcada de representar al símbolo que nos distingue como cubanos en cualquier parte: la bandera de la estrella solitaria. Los colores negro y rojo de la rama del árbol manifiestan el carácter rebelde de la isla que nunca se dobla, en una alusión a la bandera del Movimiento 26 de Julio.

Sobre la grafía y otros aspectos

- La grafía empleada en las letras del nombre de la distro comunican el destino final hacia el cual van dirigidos todos nuestros esfuerzos: la educación cubana, los niños y adolescentes de todo el país.
- El degradado de fondo en azul es solo una cuestión de enfoque del diseño desde la perspectiva de la seriedad del asunto que se trata.

Características

- Menú con carpeta de Historia (biografías de nuestros héroes, libros de historia en formato electrónico, etc)
- Multimedia del Che
- Multimedia de los símbolos patrios
- Aprovechar algunos software educativos de distros como EdUbuntu



- Carpetas en el menú con software y juegos educativos.
- Carpetas en el menú de Biología, Química, Física, Matemática, etc, con software para esas asignaturas y sitios Web que traten estos temas.
- Tema de escritorio con los colores de la bandera cubana.

Quiénes Integran el equipo

Coordinadores

Dairo Reyes Rodríguez
Eiger Mora Moredo

Colaboradores

Marcos Luis Ortiz Valmaseda
Jorge Luis Rodríguez Carpio
Oscar Luis Garcell Martínez
Félix Iván Romero Rodríguez
David Padrón Álvarez
Abel García Vitier
Jorge Luis Betancourt González
Yosep Yasmany Pérez Pérez
Edgar Pérez Nash
Yunier Vega Rodríguez
Yurisleidys Leiva Zúñiga
Raisel Milián Rodríguez
Maikel Laurencio Giralte
Alejandro Jorge Murillo Pérez
Yasmany Cubela Medina
Fernando Nápoles Gamez
Dairon Pérez Frías
Yirianni Rivero Escalona

Invitados a asesoramiento

Hector Rodríguez Figueredo
Abel Meneses Abad
Yoandy Pérez Villazón
Disnel Rodríguez Rodríguez

Tareas a realizar

Diseño Gráfico: Identidad (Grub, Bootsplash, GDM, GTK2, Iconos, Metacity, Wallpapers, Elementos del CD, etc).

Diseño del Menú: Organización de los ítems y grupos de aplicaciones

Hardware: Investigar hardware existente en centros educativos joven club

Software: Investigar software educativo que se puede utilizar (extranjero)

Desarrollo: Aplicaciones a desarrollar para la distro.

Asesoramiento Legal: Investigación de temas legales (licencias)

Asesoramiento Educativo: Investigación de contenido educativo.

Distro aprobada para su desarrollo

Ubuntu (Xubuntu + Edubuntu), Xubuntu es una distribución basada en Ubuntu GNU/Linux y ésta a su vez en Debian GNU/Linux, su curva de aprendizaje es relativamente fácil en comparación a otras metadistribuciones como Gentoo. La empresa Canonical Ltd., desarrolladora de Xubuntu, es una empresa sudafricana encabezada por el millonario Mark Shuttleworth, por lo que no se acogen al bloqueo ni a las leyes norteamericanas y además esta empresa reconoce a Cuba dentro de sus LoCoTeams.



Sistema de registro de usuarios de Tocaroro GNU/Linux

Nonacimal Tocaroro

Nuestra máxima: "Siempre somos 1 más..."

Nonacimal Tocaroro es el sistema numérico empleado para registrar a los usuarios de la distro. La cantidad de elementos de la base (9) se tomó en reconocimiento a las personas que participaron en la primera reunión de la Comunidad de la Distro y representa nuestra manera diferente de hacer las cosas.

El alfabeto de este sistema numérico no tiene al cero como elemento de manera que sus dígitos llegan efectivamente hasta el 9.

¿Por qué no usar el 0?

El cero es un símbolo del que se infiere el vacío, y nuestra comunidad, por principio, es de todo el que quiera aportar. Por otro lado, fue comenzando por 1, y no por en 0, que se decidió aleatoriamente la numeración de los colaboradores que asistieron por voluntad propia al primer encuentro, la máxima: "Siempre somos 1 más..." sintetiza esta idea, y su concepto se fundamenta en el siguiente principio matemático, que es verdadero siempre para el sistema nonacimal Tocaroro.

¿Cómo convertir de nonacimal a decimal y viceversa?

Si usted quisiera convertir un número del sistema decimal ordinario a nonacimal Tocaroro, tendría que adicionar siempre 1 a todos los valores que se obtienen del método de conversión ordinario del sistema nonacimal que comienza por el 0 y excluye al 9.

Ejemplo:

Sea el número 20 decimal:

$$20 \bmod 9 = 2;$$

$$(20/9) \bmod 9 = 2;$$

$$20/(9*9) = 0: \text{condición de parada del método de división continua.}$$

Tomamos ahora todos los restos comenzando por el último, o sea 22, y les adicionamos 1. El número finalmente obtenido sería la cifra tocaroro, es decir 33.

Ingeniería Inversa

Tomamos la cifra tocaroro 33 y le restamos 1 a todos los dígitos, es decir que nos quedamos nuevamente con 22. Ahora:

$$2*9^1 + 2*9^0 = 20$$

Como puede apreciarse en el ejemplo anterior, 1 más... es la característica que distingue a Tocaroro GNU/Linux

Conclusiones

La distribución Tocaroro GNU/Linux siempre será Software Libre, liberada bajo la licencia GNU GPL versión 2 o superiores, será entregada al Estado Cubano para que haga uso de esta en beneficio de los niños y jóvenes que hoy estudian en todos los centros educacionales del país. Este proyecto siempre va a ser un proyecto de y para las comunidades de desarrollo de la Universidad, por lo que no puede convertirse nunca en un proyecto a puertas cerradas, de lo contrario perdería el sentido para lo que ha sido creado, que no es más que una forma de producir colaborativamente. Por tanto todo que se sienta con voluntad para integrar Tocaroro GNU/Linux puede hacerlo ya sea para desarrollar o para aportar ideas.



Infodrez, el sitio Web del ajedrez de la UCI

Raciel Garrido Torres, Jose Rodríguez Alarcón, Ing. Abel Meneses Abad
rgarrido@estudiantes.uci.cu, jalarcon@estudiantes.uci.cu, abelma@uci.cu
Proyecto UNICORNIO

El desarrollo de software para el ajedrez, es una rama de la informática con años de experiencia. Hoy se conocen versiones de clientes y servidores muy avanzados con equipos enteros de desarrollo de software y planes de desarrollo muy bien elaborados. Conocemos también los esfuerzos en materia de inteligencia artificial para el desarrollo de motores de juego similares a los empleados en la Computadora DeepBlue, tales como el GNU Chess.

La UCI y el ajedrez

En la Universidad de las Ciencias Informáticas (UCI) el juego ciencia se vio favorecido con la creación de la cátedra de ajedrez, y una única experiencia, muy aislada y carente de visión a largo plazo: Ucidrez.



Figura 8. Portal de Ajedrez Infodrez

¿Era Ucidrez capaz de cumplir con las necesidades de la comunidad ajedrecística en la universidad?

Teniendo en cuenta que sirvió de base para nuevas investigaciones, Ucidrez se alejaba de cumplir con las normas y estándares internacionales para el diseño y desarrollo de aplicaciones Web, impedía que un gran número de ajedrecistas pudiera tener acceso a ella por la sencilla razón de utilizar navegadores Web que se rigieran por las normas de la W3C. Un grave problema de diseño: NO ERA ESTANDAR. Por ello, la respuesta a nuestra pregunta es no: Ucidrez no cumplía con las necesidades de la comunidad ajedrecística de la UCI.

En marzo del 2006, en la facultad 10 de la universidad se realizó la total migración hacia GNU/Linux utilizando la distribución Debian, lo que afectó el desarrollo de la comunidad ajedrecística en la facultad, pues el principal punto de contacto entre ajedrecistas de la universidad no era compatible con navegadores Web libres como Mozilla. Ucidrez no era universal.

El líder del proyecto UNICORNIO al percatarse de ello se dirige hacia la Cátedra de Ajedrez en búsqueda de una solución y descubre entonces que Ucidrez era una aplicación sin equipo de soporte técnico y además observa el descontento que producía para la cátedra por la cantidad de fallas que este tenía. Ucidrez no era un problema solo para la facultad 10 sino para todo el universo estudiantil deseoso de interactuar en el mundo del ajedrez.



Nuevo Software

El Proyecto UNICORNIOs se da a la tarea de crear un equipo de desarrollo integrado por personas que gustaran del ajedrez y desarrollaran un interés especial por la creación de una nueva aplicación, que satisficiera las necesidades de la cátedra, cumpliera con los principios del Software Libre y sirviera de base para el desarrollo colaborativo en la universidad. Este proceso dura alrededor de 5 meses debido al abandono o no idoneidad de los equipos previos. En encuentros posteriores se realiza un levantamiento de requisitos para desarrollar una versión inicial de lo que será el nuevo software: Infodrez.

Elección de las tecnologías

Teniendo como premisa la cantidad de personas que juegan ajedrez en la universidad (más de mil) y que el juego basado en tiempo requiere rapidez de conexión y reducción de tiempos de respuesta, nos dimos a la tarea de realizar una investigación sobre posibles tecnologías a utilizar, de manera que disminuyera la carga en el servidor, y reducir las transacciones de red.

Durante la investigación se encontraron algunas aplicaciones basadas en Java que requerían un cliente implementado, las cuales fueron desechadas por el gran consumo de recursos en su implementación pues no se contaba con la tecnología adecuada para su desarrollo (PCs con 256 MB RAM). Otra variante estudiada fue una aplicación Web parecida a Ucidrez pero con el total aprovechamiento de las ventajas que esta implica al utilizar como cliente navegadores Web implementados para todas las plataformas existentes. El siguiente paso fue la elección de la tecnología: ASP.NET, PHP o Perl.

No se utilizó ASP.NET pues muchas veces las aplicaciones desarrolladas en esta tecnología se tornan un poco lentas, además de no ser multiplataforma y de que solo puede ser montada en servidores Windows del tipo IIS; tampoco de trabajó con Perl, que aunque es mucho más rápida no cumplía con las expectativas, siendo hasta cierto punto difícil su implementación y no posee IDEs con las prestaciones de Kompozer, Quanta o ZendStudio. Finalmente se elige PHP. Y una base de datos MySQL era la ideal por varias razones: posee una alta velocidad de procesamiento de datos y operaciones, confiabilidad en el gestor de base de datos ligado a la rapidez de ejecución y conexiones con los scripts PHP, y por último su mayor ventaja, radica en que es un sistema multiplataforma, puede ser implementado sin importar el sistema operativo que utiliza.

Infodrez 2.0

La segunda etapa de desarrollo de la aplicación debe contar con otras funcionalidades (mejorando las que ya tiene pensadas para sustituir Ucidrez 2.0) Entre ellas:

- Premarcado
- Juegos con bonificación de tiempo
- Visualización de partidas que se están jugando por otros usuarios
- Control sobre partidas jugadas. Capacidad de exportar a PGN.
- Visor de partidas PGN



Figura 9: Juego Online incluido en Infodrez

Infodrez 3, 4...

Se tiene pensado desarrollar otros módulos de esta aplicación para los diferentes niveles de enseñanza, lo que está hecho hasta el momento está dirigido a personas con un nivel medio o superior, que no es el caso de un niño de una escuela primaria o de alguien que nunca ha jugado ajedrez, para ellos será necesaria una plataforma mucho más básica y didáctica, la idea es sin embargo, hacerlo extensivo a todos y que sirva como una herramienta más dentro del aprendizaje del ajedrez de cada cubano, y lograr así que sea un producto informático de carácter nacional.



Figura 10: Logotipo temporal de Infodrez

Conclusiones

Hasta el momento se ha cumplido con el compromiso inicial: *la migración del Portal Ucidrez a plataformas libres*. La seriedad del trabajo, la constancia y la permanente comunicación, consejos que proponemos a cualquier desarrollador de software, han ocasionado que además de una aplicación robusta, amigable y portable, simultáneamente se ha favorecido el uso y crecimiento de una comunidad con fines científicos: la comunidad de ajedrez. Sólo resta invitarlo, si es estudiante o trabajador de nuestra universidad, a que visite nuestro portal¹³, o puede visitar nuestra universidad para ver materializado todo lo explicado. Infodrez 1.0 es libre y puede utilizarse su código.

¹³ <http://infodrez.uci.cu>



Autenticación de Plone contra el servidor de LDAP de la UCI

Keiver Hernández Fernández
khfernandez@estudiantes.uci.cu
Proyecto UNICORNIOS

Plone es un sistema de administración de contenido (CMS) amigable al usuario, sumamente potente y flexible. Por su parte LDAP (Lightweight Directory Access Protocol) es un protocolo de red que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red. El CMS Plone permite la integraciones con servidores LDAP para la autenticación de usuarios de forma fácil y eficiente. Para lograr la integración de Plone con el servidor LDAP, se necesitan los productos de Plone, LDAPUserFolder y LDAPMultiPlugins así como el paquete de Python, python-ldap.

Una vez instalados los paquetes, es necesario crear un objeto de tipo LDAPMultiPlugins dentro del objeto `acl_users`. Estos son los datos para configurar en el objeto LDAPMultiPlugins para permitir el acceso de los usuarios del dominio UCI al portal.

LDAP Server:	uci.cu
Read Only:	[Marcado]
User Base DN:	OU=UCI Domain Users, DC=uci, DC=cu
Login Name Attribute:	Usuario Dominio UCI (mailNickname)
User ID Attribute:	Usuario Dominio UCI (mailNickname)
RDN Attribute:	Nombre Completo (displayName)
Group Storage:	Groups not stored on LDAP server
Manager DN:	CN=nombre completo, OU=F10, OU=Students, OU=UCI Domain Users, DC=uci, DC=cu
Password:	[contraseña del dominio del usuario especificado]
Default User Roles:	Member
User object classes:	top, person, organizationalPerson, user

Ahora se definen las propiedades del LDAP que tendrán los usuarios en Plone, esto se hace en la pestaña *LDAP Schema*, en la sección *Add LDAP schema item* y se adicionan las propiedades con los siguientes valores:

Title	Cargo en la UCI	description
Mail	Correo UCI	email
Department	Lugar o Grado de Usuario	location
DisplayName	Nombre Completo	fullname
MailNickname	Usuario Dominio UCI	userid

La casilla *Multi-valued*, no se marca en ningún caso. Se obtendrán mejores resultados si se borra el Objeto *mutable_properties*, en el objeto `acl_users`. Terminado de hacer estas configuraciones el sitio Plone está listo para autenticar los usuarios del Dominio UCI.

Enlaces de interés:

Artículo en la WikiProd

http://wiki.prod.uci.cu/index.php/Autenticacion_de_Plone_&_LDAP

Productos en el servidor de Documentación

<http://10.128.50.121/Software/Instaladores/Plone>

Creación de un repositorio local¹⁴

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

Este artículo constituye una guía simple que permitirá crear un repositorio local, para ello primero debe tenerse instalado el sistema con todo lo que va a ser utilizado en el repositorio.

Estructura del repositorio

```

Carpeta_acceso (CA) = debian, debian-security, alternativo...
                        Ubuntu=ubuntu,ubuntu-security,ubuntu-proposed...
..dists
....distribucion = Debian= stable(etch), testing(lenny), unstable(sid)
                        Ubuntu= dapper, edgy, feisty
.....seccion = Debian= main, contrib, non-free
                        Ubuntu= main, universe, multiverse, restricted
.....binary-i386
..pool

```

Construyendo la estructura

Debe estar ubicado en la ruta /home/user/Desktop/CD. La carpeta CD debe estar creada en el escritorio.

```
[user] [~/Desktop/CD]$ install d debian/dists/etch/main/binary-i386/
[user] [~/Desktop/CD]$ mkdir debian/pool
[user] [~/Desktop/CD]$ ln -s debian/dists/etch debian/dists/stable
```

Se copian los paquetes de la caché del sistema:

```
[user] [~/Desktop/CD]$ cp /var/cache/apt/archives/*.deb debian/pool
```

Hacer el scanpackages (para ello se necesita el paquete *dpkg-dev*)

```
[user] [~/Desktop/CD/debian]$ dpkg-scanpackages pool/ /dev/null | gzip > dists/etch/main/binary-i386/Packages.gz
```

Añadir el info para el montaje automático:

```
[user] [~/Desktop/CD]$ mkdir .disk
[user] [~/Desktop/CD]$ echo "Debian Etch by eiger" > .disk/info
```

Construir el ISO, incluyendo todo lo que está dentro de la carpeta CD:

```
[user] [~/Desktop]$ mkisofs -f -J -R -o cd_repo.iso CD/ 2>salida-error.log
```

Ya está construido el repositorio local

¹⁴ http://10.128.50.121/Documentacion/Generales/Crear_Repositorios/



Las patentes de software, un problema legal

Rodolfo González Pelegrino, Angel Guillermo Borjas Almaguer
rpelegrino@estudiantes.uci.cu, agborjas@estudiantes.uci.cu
Proyecto UNICORNIOS

Una patente es un conjunto de derechos exclusivos garantizados por un gobierno o autoridad al inventor de un nuevo producto (material o inmaterial) susceptible de ser explotado industrialmente para el bien del solicitante de dicha invención por un espacio limitado de tiempo.

Las patentes se introdujeron en el siglo XVII en Inglaterra y otras partes del mundo para fomentar el avance de la ciencia. Si no se protegían las ideas, los inventores las habrían mantenido en secreto y nunca hubieran llegado al público. Sin embargo las patentes lejos de proteger invenciones ingeniosas constituyen bloques de segmentos, cada patente es un medio para impedir por muchos años que otros utilicen ciertos conocimientos.

Es un error pensar el hecho de que son otorgadas por invenciones ingeniosas. Incluso la "mafia de las patentes" admite que estas ya no protegen los grandes adelantos. Son los pequeños pasos los que son patentados en estos días. Por ejemplo, sólo Siemens tiene aproximadamente 50000 patentes. No es posible que todas sean invenciones brillantes. La inflación de patentes es un problema global, ya que millones de patentes son válidas en este momento, y se solicitan aproximadamente 700.000 nuevas patentes cada año. Solamente la Oficina Europea de Patentes recibe más de 150.000 solicitudes por año.

En más de la mitad de esos casos, se expide una patente. La mayor área de crecimiento de todas es la de las patentes de software. A un programador le es imposible comprobar que no infrinja ninguna patente. El Software es muy incremental y cada programa utiliza muchos conceptos. La posibilidad de patentar algoritmos y estructuras de datos daría a las compañías privadas control sobre demasiados recursos de la sociedad de la información. Gracias a patentes sobre formatos de datos (o software para leer y escribir éstos formatos) las multinacionales podrían impedir a los autores crear o distribuir arte digital de todo tipo.

Hoy en día, ya hay patentes que afectan los dos formatos de imagen más usados en la Web (JPEG y GIF), lo mismo pasa con audio (MP3) o video (MPEG, ASF). Estos y otros poseedores de patentes pueden censurar o poner peajes a gran parte del contenido multimedia del mundo. El término "mafia de las patentes" no está destinado a insultar a ninguna persona o institución en particular. Es una alusión a las deficiencias estructurales del sistema de patentes y a la creciente conducta no ética de algunas personas.

Es imposible implementar un sistema que no produzca malas patentes

Los examinadores de patentes tienen demasiado trabajo, y poco tiempo para buscar lo que se denomina 'arte previo'. Una patente no sólo tiene que ser 'no obvia', sino también novedosa; el 'arte previo' son ejemplos anteriores del invento propuesto para patente.

Se decía de Von Neumann, el gran matemático húngaro cuya contribución en el proyecto Manhattan y con la NASA dio a los Estados Unidos la bomba atómica y el viaje a la Luna, que era la única persona viva que tenía conocimiento de todas las ramas de la matemática.

En la actualidad es posible que sólo Donald Knuth, autor de "The Art of Computer Programming" y apodado cariñosamente "Dios" en círculos de programadores, tenga todos los conocimientos para saber si un algoritmo o programa es novedoso o no.

Es casi imposible asegurar que no se infringe alguna patente

El hecho de que las patentes de software estén redactadas en términos tan oscuros, el alto precio de investigar sobre las patentes existentes (la principal fuente de ingresos de la EPO son los informes sobre las patentes que hay en un determinado campo de la industria) y la gran cantidad de patentes 'obvias' impiden comprobar adecuadamente si uno está infringiendo una patente al escribir un programa, o no.



Por esta razón las patentes de software podrían suponer un freno a la industria del software, y bloquear la innovación al no impedir que un programa novedoso use código patentado en alguna de sus operaciones más triviales. La promoción del software libre es absolutamente incompatible con la introducción de las patentes de software, y no es por casualidad que la comunidad del software libre sea una de las más activas en la lucha contra las patentes de software.

Por otro lado cualquier área económica que pueda mantenerse libre de patentes de software será un buen lugar para establecer negocios basados en software libre. Esta ventaja competitiva no será despreciable si la industria del software libre alcanza el potencial que muchos expertos esperan.



NOVA GNU/Linux

Michel Hernández Rodríguez
mhrodriguez@estudiantes.uci.cu
Proyecto NOVA

El Software libre es el software que, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente. El mismo suele estar disponible gratuitamente en Internet, o a precio del coste de la distribución a través de otros medios; sin embargo no es obligatorio que sea así y, aunque conserve su carácter de libre, puede ser vendido comercialmente.

De acuerdo con tal definición, el software es "libre" si garantiza las siguientes libertades:

- "libertad 0", ejecutar el programa con cualquier propósito (privado, educativo, público, comercial, etc.)
- "libertad 1", estudiar y modificar el programa (para lo cuál es necesario poder acceder al código fuente)
- "libertad 2", copiar el programa de manera que se pueda ayudar al vecino o a cualquiera
- "libertad 3", mejorar el programa, y hacer públicas las mejoras, de forma que se beneficie toda la comunidad.

Es importante señalar que las libertades 1 y 3 obligan a que se tenga acceso al código fuente. La "libertad 2" hace referencia a la libertad de modificar y redistribuir el software libremente licenciado bajo algún tipo de licencia de software libre que beneficie a la comunidad.

¿Qué es una distribución?

Una distribución es un conjunto de aplicaciones reunidas por un grupo, empresa o persona para permitir instalar fácilmente un sistema operativo. En general se destacan por las herramientas para configuración y sistemas de paquetes de software a instalar.

Las herramientas que suelen incluirse en las distribuciones de este sistema operativo se obtienen de diversas fuentes, incluyendo de manera importante proyectos de código abierto o libre, como el GNU y el BSD.

La distribución Nova. Su origen

Fue un proyecto pensado y creado desde sus inicios por 2 estudiantes de 5to año pertenecientes a la Universidad de las Ciencias Informáticas, donde en la actualidad ejercen como profesores cumpliendo de esta forma con su servicio social, cuyos nombres son *Anielkis Herrera González* y *Yoandy Rodríguez Martínez*, quienes llegaron a concretar muy certeramente sus ideas con la creación del proyecto NOVA, en dicha institución.

Nova es una distribución que hereda de Gentoo/Linux los componentes base de su arquitectura. El sistema de selección y la filosofía de desarrollo han sido asimilados y transformados en herramientas con una mejor integración con el escritorio. A pesar de la variedad de distribuciones de Linux en existencia, pocas de ellas implican un proceso real de desarrollo u optimización sino que consisten en el ensamblado de paquetes de una distribución base en forma de solución o muestra, y la mayoría no permite optimizaciones o personalizaciones para entornos específicos o sistemas dedicados.

Una distribución desarrollada en nuestro país no sólo podría ser orientada y optimizada según nuestras necesidades de migración y desarrollo de software sino que también proporcionaría una plataforma para la investigación y la producción a pequeña, mediana y gran escala.

Uno de los principales objetivos de este proyecto, es desarrollar una distribución de GNU/Linux adaptada a las necesidades de migración, servicios y aplicaciones de nuestro país y clientes. Siempre teniendo en cuenta una estructura de desarrollo de versiones justamente a la medida, según varíen las necesidades originales de la migración y clientes.



Figura 11: Logo de NOVA
GNU/Linux

¿Por qué desde Gentoo-Linux?

Gentoo Linux es un producto que gana usuarios aceleradamente en todo el mundo. Es una distribución especial que aunque no esta pensada para usuarios noveles puede adaptarse a las necesidades específicas de cualquiera. Gracias a una tecnología denominada Portage, Gentoo Linux puede convertirse en un servidor ideal, estación de trabajo para desarrollo, escritorio profesional, sistema para juegos, o solución para cualquier otro uso. Dada su casi ilimitada adaptabilidad, denominamos a Gentoo Linux como una metadistribución.

Desarrollo de utilidades

La configuración y administración de un sistema Linux es usualmente un proceso complicado que implica el conocimiento acerca de varios archivos o procedimientos que normalmente escapan al interés del usuario común. El directorio `/etc/`, donde radica los archivos de configuración de cada programa instalado en el sistema, cuenta con aproximadamente 2190 archivos en un sistema típico, de estos cerca de 2179 están relacionados con los parámetros de funcionamiento de algún programa.

Estos archivos a pesar de estar mayormente en texto plano, cuentan una gran variedad de formatos y reglas que en caso de violarse podrían alterar la estabilidad del sistema. En el caso de Gentoo Linux existen varios scripts que se ejecutan automáticamente al instalar un nuevo software que eliminan una pequeña parte de esta barrera, pero aún así el usuario tiene que recordar nombres y modos de invocación para cada uno.

La solución Nova LNX a este problema consiste en la introducción de software utilitario que englobe los aspectos de configuración, instalación de aplicaciones y mantenimiento del sistema. Estas utilidades además se integran al modelo de seguridad. Actualmente se han desarrollado herramientas para la instalación del sistema, la configuración de carpetas compartidas utilizando el protocolo SMB, un panel de control central de aplicaciones, un buscador de informaciones sobre programas instalados en la distribución, un instalador de la distro y la instalación de nuevos paquetes entre otros.

LiveCD

Este se ha ido perfeccionando en la medida en que se han desarrollado algunos aspectos muy importantes para toda distribución, como es el caso de la creación de los LiveCD. Un LiveCD es básicamente un sistema que se ejecuta en un entorno restringido el medio físico y es de solo lectura (a diferencia de otras tecnologías como el LiveUSB donde el soporte es de lectura escritura) y que se apoya principalmente en tecnologías del BIOS que permiten elegir al dispositivo CDROM como primario para el encendido del PC. Debido a que este tipo de sistema no puede crear archivos temporales en el medio físico se utilizan puntos de montaje creados con cualquiera de los sistemas de archivos en memoria de Linux (ramfs o tmpfs) siendo el último el más utilizado al no tener restricciones acerca de la cantidad de memoria a la que puede acceder.

El mantenimiento

El mantenimiento de un producto es parte fundamental de su ciclo de producción. En el caso de Nova el mantenimiento y creación de nuevas versiones es efectuado a partir de los reportes de error de Gentoo-Linux y la retroalimentación de los usuarios. Las facilidades del portage de señalar paquetes como inestables o potencialmente peligrosos permite organizar mejor el trabajo y generar parches al código o personalización defectuosa.



El servicio de atención al usuario incluye:

- Actualizaciones del portage.
- Avisos utilizando el sistema GLSA.
- Actualizaciones del repositorio e inclusión de nuevo software.



Metodología para la Migración a Software Libre (Parte I)

Ramón Paumier Samón
rpaumier@estudiantes.uci.cu
Proyecto UNICORNOS

Una metodología para la migración a Sistemas de Software Libre, es un extenso documento de unas 200-300 páginas en formato de PDF, lo que hace molesto su estudio y difícil su comprensión. Si Ud. es de los que han comprendido la necesidad de la migración y se han iniciado en ese largo y complejo proceso, entonces sabrá que esto que le digo no es nada nuevo, y ese es uno de los primeros contratiempos a los que se enfrenta.

Los deseos de elaborar un proceso de migración acorde a las necesidades actuales y, por sobre todas las cosas, “propio”, obligaron a la búsqueda de intentos de migración en una amplia gama de países. Finalmente, 5 de ellos fueron escogidos, considerando los importantes aportes y las interesantes soluciones que ofrecían ante complejas situaciones: las directrices de la Unión Europea, el Plan Nacional para la Migración de Venezuela, el Plan Nacional para migrar las Entidades Públicas de Perú, la Guía de la Universidad de los Andes en Venezuela y la de la Universidad de Misiones, en Argentina.

Fruto de una investigación de varios meses, como parte del Trabajo de Diploma del autor de este artículo, ahora es posible comprender mejor y de una manera más ilustrativa qué, con quién, cuándo y cómo hacer efectivo un empeño de tal magnitud. Les propongo en esta ocasión, un acercamiento a la primera de las 3 etapas que componen la metodología.

Etapas 1: Preparación

Las directrices europeas insisten en definir el proyecto desde el principio con la mayor cantidad de detalles posibles. Optan por la justificación temprana del por qué migrar y prueban a menor escala las fases del proyecto una vez definido. Sin embargo, consideran la formación de los usuarios como algo trivial que debe implementarse una vez que se haya migrado.

Perú aboga como parte de la preparación, por la sensibilización institucional respecto a la migración a Software Libre y la organización institucional para la implementación del Software Libre, lo que se considera indispensable para emprender cualquier intento de este tipo. Proponen la elaboración, de conjunto con la administración de la institución, el plan de migración e instan a esta, a elaborar un plan de acción institucional para apoyar la migración y exigen se chequee su cumplimiento. Esta guía alerta sobre la necesidad de un levantamiento informático, de hardware, software e intelectual, para evaluar el grado de factibilidad de la migración y el conocimiento especializado del personal de informática existente en la institución, que será el encargado del desarrollo del proceso. Se determina la mejor alternativa para migrar en base a la tecnología, los servicios que se brindan y el presupuesto del que se dispone.

A diferencia de los documentos anteriores, que intentan minimizar el proceso de migración y hacerlo en el óptimo y/o menor tiempo posible, el Plan Nacional de Migración de Venezuela propone más bien un proceso a largo plazo y en el que definitivamente habrá que hacer determinados ajustes, aunque, sin dejar lugar a dudas, este constituye un claro ejemplo de los sólidos pasos del proceso revolucionario venezolano. Esta propuesta centra su etapa de preparación en la introducción de la filosofía del Software Libre tanto en el sistema educacional como en la administración pública, en el lanzamiento de campañas de divulgación del mismo y la firma de convenios para garantizar todo lo anterior. Proponen la creación y puesta en funcionamiento de un portal para dar información y soporte a los usuarios.

Resulta interesante el aporte que realiza la guía de migración de la Universidad de Misiones que propone el dictado de cursos a los administrativos de la institución, con el fin de actualizarlos y tener un contacto más estrecho con los programas que utilizan diariamente y al personal en funciones técnicas o especialistas informáticos, con el fin de capacitarlos. Así como, la introducción de determinadas herramientas en el ámbito cotidiano como el caso de navegadores y paquetes de ofimática, para ir familiarizando a los usuarios y no exista mucha resistencia al cambio.



Basado en estos análisis, se arriba a la conclusión, de que un futuro proceso de migración de software propietario a Software Libre, deberá desarrollar durante su etapa de preparación, el cumplimiento de las siguientes fases:

La justificación de la migración, fase importante que definirá el grado de comprometimiento de los usuarios con la necesidad del proceso y donde se realizará:

- La evaluación de los costos de uso de una u otra plataforma. Incluso podrán evaluarse los costos de acometer la migración,
- La evaluación de las aplicaciones, buscando el equivalente óptimo a las aplicaciones propietarias,
- El análisis de las ventajas del uso de esta nueva plataforma libre.

La planeación de la migración, que incluirá:

- La realización del levantamiento informático con vistas a prevenir incongruencias de hardware y software, además de evaluar el capital humano del que dispone la institución para la realización de la migración,
- La elaboración del Plan de Migración a Software Libre por parte de los especialistas encargados,
- La sensibilización de la institución con la realización del proceso y la organización de la misma para acometerlo, mediante la elaboración conjunta del plan de acción institucional que permitirá seguir de cerca y más que eso, garantizar el desarrollo exitoso del proceso.
- La formación y capacitación de la mayor cantidad de personal posible para intervenir en la realización del proceso y acometer labores de soporte, durante la migración y posterior a esta.

Las pruebas pilotos al diseño, momento indicado para:

- Valorar el Plan de acción de la institución.
- Probar en tiempo real todo lo planificado.

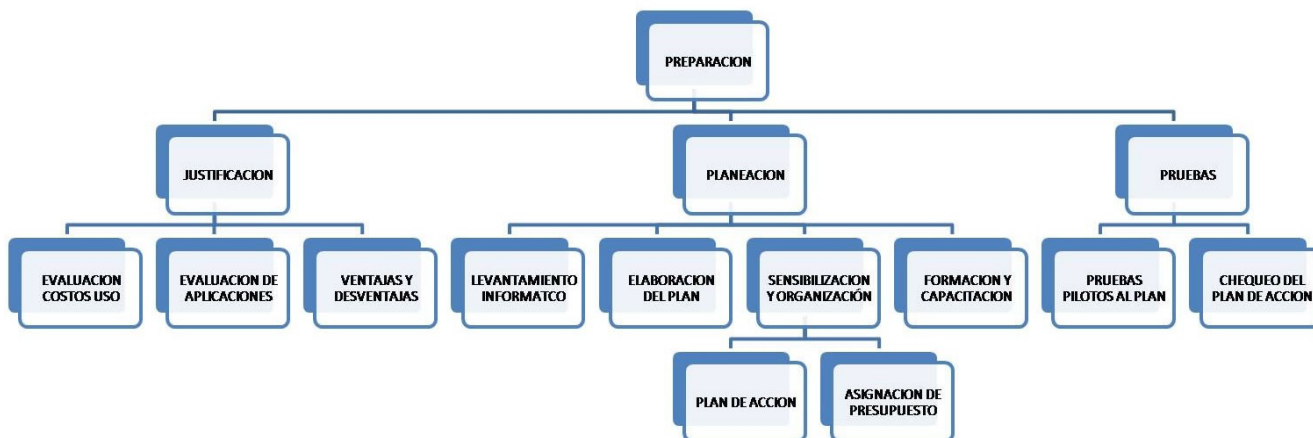


Figura 12: Esquema de la etapa de preparación



Entorno de Desarrollo para Python

Daniel Hernández Bähr, Noel González Fernández
dbahr@estudiantes.uci.cu, ngonzalezf@estudiantes.uci.cu
Proyecto NOVA

Cuando, a principios de primer año, me dijeron que el lenguaje que estudiaría era Python, pensé que terminaría dedicándome a hacer páginas Web; no podía estar más equivocado. Hoy por hoy, desarrollo todas mis aplicaciones de escritorio en Python, y una de ellas es el centro de este artículo.

¿Qué es?

PyDE (Python Development Environment por sus siglas en inglés) es, como su nombre lo indica, un entorno de desarrollo hecho por y para Python. La interfaz gráfica de usuario está diseñada en la versión 2.12.1 de Glade y se *parsea* utilizando el *binding* PyGtk de Python para Gtk. El código se escribió en el editor SciTe en su versión 1.71. Está diseñado para correr sobre un sistema GNU/Linux con escritorio Gnome, pero por las características de portabilidad del lenguaje y las librerías usadas solo requiere el gestor de ventanas xorg.

¿Por qué hacerlo?

Hubo 2 motivos que nos impulsaron al desarrollo de PyDE: consolidar los conocimientos que adquirimos en nuestro primer año en la Universidad en las asignaturas *Introducción a la Programación* y *Programación I*, y la carencia de un entorno de desarrollo libre para Python.

¿Qué hace?

Aunque aún se encuentra en una etapa muy temprana de su desarrollo ya cuenta con algunas de las funcionalidades básicas inherentes a todo entorno de desarrollo:

- Creación y edición de proyectos
- Trabajo con varios módulos en un mismo proyecto.
- Coloreado de sintaxis (incluye, Números de línea y resaltado de la línea actual).
- Completamiento de código (en una versión muy básica aún).
- Ejecución de código en consola

Es válido, asimismo, recalcar que el módulo utilizado en la simulación de la consola no está incluido por defecto en la mayoría de las distribuciones más utilizadas, quedando de esta manera como una dependencia.

¿Qué falta por hacer?

Aunque se ha hecho bastante, aún queda mucho trabajo por delante. En el momento en que se redacta este documento nos encontramos trabajando en el completamiento de código y el diseñador para interfaces gráficas de usuario con **Gtk**. Pensamos también, paulatinamente, incorporar otros lenguajes, principalmente **Perl** y **Bash**.

Completamiento de código (en una versión más potente):

Su función es facilitar al usuario la creación del código, reconociendo en todo momento qué elementos intenta usar y mostrárselos de forma tal que pueda seleccionar un identificador de una cantidad de caracteres relativamente larga con un *click* o usando unos pocos caracteres. Se divide en 4 problemas fundamentales:

- Definir qué palabra está tecleando el usuario. Para ello utilizamos una expresión regular sencilla que determina si el carácter entrado por el usuario es alfanumérico o un subrayado (" _") y buscamos hacia el principio de la palabra mientras el carácter anterior cumpla con la misma condición.
- Conocer en todo momento los identificadores que está utilizando el usuario en su proyecto.
- Si el carácter introducido no cumple con la condición del punto anterior entonces buscamos el texto de todos los módulos que no estén actualizados -por no actualizado se entiende que se halla hecho algún



cambio y no halla sido registrado- y se construye un diccionario o *tabla hash* para cada módulo con todos los identificadores declarados en él. Por último se actualizan los módulos no actualizados de forma tal que no sea necesario buscar en todos los módulos cada vez que se va a construir las tablas.

- Encontrar todos los identificadores que comienzan con la combinación de caracteres que está entrando el usuario. Se busca primeramente en la tabla que contiene los identificadores de todos los módulos, de no coincidir el identificador actual con el nombre de ninguno de ellos se supone que el identificador que se desea usar se encuentra en el módulo actual, apto seguido se procede a buscarlo en la tabla correspondiente al módulo actual y se obtiene una lista con todos los identificadores que comienzan de esa manera. En el caso de que el identificador coincida con el nombre de un módulo, se procede a obtener una lista con todos los identificadores registrados en él. Mostrar un menú, lo más cerca posible del punto donde se encuentra el cursor -pero sin obstruir la visión del texto que se escribe-, con todas las posibles elecciones del usuario.

Diseñador de interfaces:

Su función consiste en acortar el tiempo de diseño de la interfaz, agilizando el proceso y haciéndolo más amigable para el programador. Aunque existen algunos diseñadores de *GUIs* que pueden usarse en conjunto con python estos no cumplen a cabalidad con la intención de nuestro trabajo, pues o no generan el código Python (sino un **XML**) o son muy pesados para correrlos en PCs de bajas prestaciones.

Debugger:

Permite al programador conocer en qué punto del programa se encuentra un error de su código.

¿Cómo es?

PyDE, en su versión actual, está formado por varias ventanas, la mayoría de las cuales el usuario puede elegir si desea utilizarlas o no. Ellas son:

Principal, es el núcleo de PyDE, y desde ella se realizan las principales acciones, desde la creación de proyectos hasta la ejecución de los mismos. Está compuesta por: *Barra de Menús Barra de Herramientas*

Código, su función es editar el proyecto, con todas las funcionalidades que necesita. Está compuesta por: *Barra de Herramientas Cuaderno Una Vista de Texto* por cada módulo en uso por el usuario.

Debug, muestra el proceso de ejecución del *script* y muestra los errores que este pueda acarrear. Está compuesta por: *Cuaderno Una consola embebida para la ejecución. Una vista de texto para mostrar los posibles errores.*

Mapa, muestra todos los módulos existentes en la carpeta del proyecto. Está compuesta por: *Vista de Árbol*

Metas

La principal meta de nuestro trabajo es promover el uso de Python para aplicaciones de escritorio en sistemas GNU/Linux y que PyDE pueda ser utilizado como plataforma docente para impartir las asignaturas de *Introducción a la Programación* y *Programación I* en cursos venideros en la facultad X y más adelante en cualquier facultad de la universidad.

El proyecto NOVA piensa, una vez terminado, utilizarlo como entorno oficial de desarrollo para las aplicaciones desarrolladas en Python para la distribución. PyDE cuenta con un equipo de desarrollo bastante reducido, pero nuestro interés es que toda la comunidad se integre, si lo desea, al desarrollo del mismo. Así mismo estamos buscando un nombre más identificativo con nuestra idiosincrasia, y por ahora estamos valorando llamar *Majá* a la versión definitiva. Nos gustaría conocer sus opiniones al respecto.

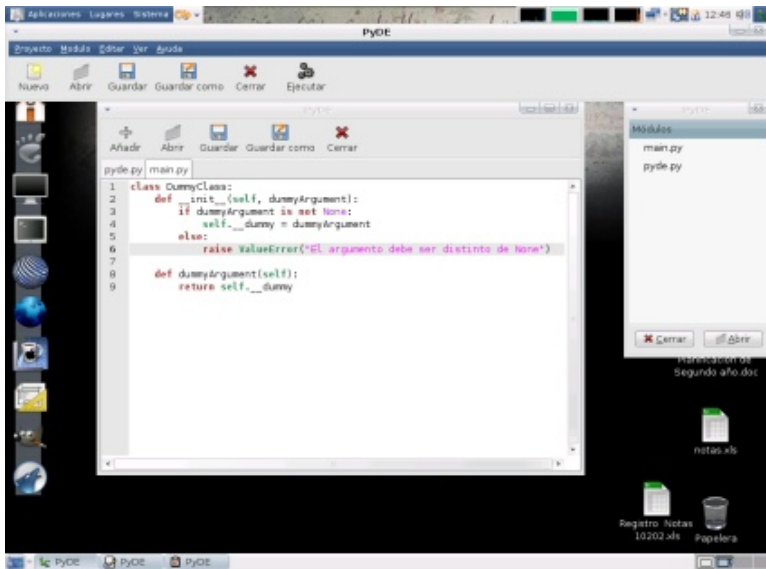


Figura 13: Captura de pantalla que muestra el IDE



Concurso de Software Libre de la UCI

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

A inicios del año 2006 y en tierras lejanas, donde el Software Libre cobra fuerza y muchas personas trabajan por mantener dicha filosofía, se lanzó el 1er Concurso Universitario de Software Libre en España. Minutos más tarde y a través de las ya originales alertas de Google la noticia del evento se paseó por el mundo; al poco rato a alguien se le ocurrió la idea de la realización del concurso aquí en la UCI y desde ese momento comenzamos a soñar nuestro 1er Concurso Universitario de Software Libre.

Concluyó el pasado curso y en la etapa de las BET, con gran cantidad de tareas como la realización del primer curso de GNU/Linux para los IPI del país y la grabación de las primeras teleclases, se comenzaron a escribir las primeras ideas y bases de nuestro concurso.

Al iniciarse el curso escolar 2006-2007 se comenzó con el estudio de la propuesta inicial, se valoraron ideas nuevas en la reunión de la Comunidad de Software Libre de la UCI y comenzamos a trabajar verdaderamente en el fin propuesto, estudiamos los distintos Weblogs de internet para elegir cuál utilizaríamos para montar los trabajos del concurso, utilizando las ventajas del Software Libre tomamos los logos del concurso de Software Libre lanzado en España para adaptarlos al nuestro y realizar nuestra propia versión, y montamos entonces todo el sistema para la realización del mismo, el día 23 de marzo durante una reunión mensual de la Comunidad de Software Libre de la UCI lanzamos la versión 0.1 de las bases que ya se habían redactado y el día 4 de abril se abrió la fase de inscripción de los trabajos, el día 23 de abril cerraba entonces la fase de inscripción y todos los trabajos inscritos que fueron aprobados entonces comenzaban la realización e implementación de sus proyectos. Al cerrar las inscripciones contábamos entonces con un total de 16 trabajos, algunos de gran calidad y utilidad práctica para los usuarios de GNU/Linux y otros de gran información acerca de lo que significaría una migración hacia otra plataforma, los trabajos presentados se exponen en la tabla.



NOMBRES	FACULTAD	TÍTULO
<u>Yurileydis Leiva Zuñiga</u> <u>Dairo Reyes Rodriguez</u>	3	Estudio de costos por conceptos de software propietario y propuesta de migración.
<u>Norge Fajardo Vega</u>	4	Aplicación para cambiar la contraseña local.
<u>Dairon Pérez Frías</u> <u>Oscar Luis Garcell Martínez</u>	4	Generador de componentes para <u>Zope 3</u>
<u>Irina Elena Argota Vega</u> <u>Raúl Pérez-alejo Neyra</u>	5	Guía para el desarrollo de interfaces gráficas para <u>gnome</u> .
<u>Leonel Zalazar Videaux</u>	5	<u>SmbConfig</u>
<u>David Padrón Álvarez</u>	6	Diseño gráfico con Software Libre , una alternativa real.
<u>José Jorge Lorenzo Vila</u> <u>Dariem Pérez Herrera</u>	8	<u>Ecumenix v0.1</u>
<u>Jorge Antonio Díaz</u> <u>Humerto Salazar</u> <u>Fabian Finalé</u>	8	SPX-AS
<u>Yunier Vega Rodriguez</u>	10	Tema de GDM para los laboratorios de la facultad X
<u>Jorge Luis Carpio Rodríguez</u>	10	<u>Quipus</u> , Sistema de instalación.
<u>Dairon Pérez Roldán</u>	10	Sistema de Clonación de Imágenes
<u>Oscar Andres Casas Machado</u> <u>Marcel R. Sanchez</u>	10	<u>gtcopyx</u>
<u>Eiger Mora Moredo</u>	10	Sistema para configuración del <u>evolution</u> en la UCI
<u>Román Miguel Valdivia Renó</u>	10	Software Libre , alternativa medioambiental

Tabla 6. Trabajos presentados en el Concurso de Software Libre de la UCI

El día 23 de mayo en la noche celebrábamos el concurso en el docente 3 (chino), con la participación de 12 de los trabajos inscritos inicialmente; los resultados aún no los tenemos, pero sabemos de antemano que el concurso cumplió las expectativas: todos aprendimos un poco más sobre el Software Libre y pusimos a disposición de los demás nuestros conocimientos de una manera colaborativa, como nos ha enseñado su filosofía de trabajo en equipo.

Desde ya, felicitaciones a los participantes y nos vemos el próximo curso con la convocatoria a la 2da edición de este concurso.



Tiflotecnología: accesibilidad en GNU/Linux

Marcos Luis Ortiz Valmaseda
mlortiz@estudiantes.uci.cu
Proyecto UNICORNIOS

El proceso de migración a sistemas operativos libres siempre constituye un problema muy complejo, condicionado no sólo por el tipo de usuarios al que va destinado dicho sistema, sino por la búsqueda de software alternativo a los de sistemas privativos como Microsoft Windows y Mac. Y se hace más difícil cuando el usuario final del sistema es una persona con discapacidades físico-motoras, visuales o de cualquier otro tipo.

Pero, esta realidad está cambiando de tono con la revolución de los sistemas operativos, al disponer de mejores alternativas en este sentido en el amplio mundo del Software Libre. Tan así está el desarrollo de este tipo de software que han surgido nuevas tecnologías para darle solución a estos problemas. Un ejemplo claro es la *tiflotecnología*.

Ustedes se preguntarán: ¿qué es la tiflotecnología? Esta palabra proviene del griego tiflo, que significa ciego. Se designa como el conjunto de ayudas técnicas destinadas a que las personas con deficiencias visuales consigan una mayor calidad de vida, centrada en una mayor autonomía personal y facilidad para el desenvolvimiento en su vida diaria.

La sociedad en que nos encontramos hoy en día ha sido calificada como la sociedad de la información y el conocimiento, apoyada de forma preponderante en las denominadas Tecnologías de la Información y la Comunicación. En este contexto, el desarrollo personal y social de sus miembros va a estar determinado en gran medida por su calificación en el manejo de estas tecnologías. Sin embargo, no todos los dispositivos tecnológicos existentes en el mercado son susceptibles a ser utilizados fácilmente por cualquier individuo. En muchos casos se requiere su adaptación para evitar que supongan un factor más de desventaja o segregación para determinados ciudadanos. La tiflotecnología, por tanto, es la adaptación y accesibilidad de las tecnologías de la información y comunicación para su utilización y aprovechamiento por parte de las personas con ceguera y deficiencia visual.

En el mundo hay varias organizaciones dedicadas al estudio de la tiflotecnología, pero en idioma castellano, la organización rectora en España es el *CIDAT*, el *Centro de Investigación, Desarrollo y Aplicación Tiflotecnológica*, creado por la ONCE en 1985 con el nombre de UNIDAD TIFLOTECNICA (UTT), y que en el 2000 cambió su denominación, cuyo objetivo principal es la prestación de servicios para todos los ciegos y deficientes visuales, abarcando los ámbitos de:

- Investigación y Desarrollo
- Evaluación
- Producción
- Distribución y Comercialización
- Mantenimiento y Reparación
- Formación de Profesionales
- Asesoramiento

En Cuba actualmente existe muy poco desarrollo relacionado con esta tecnología, tema muy amplio y al que se le puede sacar mucho provecho por las soluciones que brinda. La idea es crear un Grupo de Estudio y Desarrollo de tiflotecnología integrado como una línea de investigación dentro del Polo Científico de Software Libre y así tener una mejor base para el proceso de migración del MINED, que viene ya tocando nuestras puertas y para el cual debemos tener soluciones concretas.



Ya el Proyecto UNICORNIOS¹⁵ está trabajando para ofrecer soluciones concretas con un software sintetizador de voz llamado esSpeak¹⁶ que alcanza un buen estado de desarrollo, pero tiene algunas desventajas, como que la voz en castellano suena algo robótica, y no tiene un buen soporte para nuestro idioma, lo cual constituye un reto en el afán de elevar la calidad de vida de estas personas discapacitadas.

¹⁵ Proyecto UNICORNIOS: Servicios Especializados para la migración a Software Libre
http://wiki.prod.uci.cu/index.php/Proyecto_UNICORNIOS

¹⁶ <http://espeak.sourceforge.net>



Subversion y Apache 2

Alexander Martínez Fajardo
afajardo@estudiantes.uci.cu
Proyecto UNICORNIOS

Subversión es un sistema de control de versiones que administra el acceso a un conjunto de ficheros, y mantiene un historial de los cambios realizados. Es útil para guardar cualquier documento que cambie con frecuencia, como una novela, o el código fuente de un programa. La configuración más flexible de todas las instalaciones de servidor posibles para Subversion es la que se basa en Apache. Aunque es un poco más complicada de preparar, ofrece beneficios que otros servidores no pueden dar:

WebDAV

El servidor de Subversion basado en Apache utiliza el protocolo WebDAV que se utiliza por muchos otros programas. Esto es útil para el protocolo de red en la comunicación del cliente con el repositorio de subversión.

Arquitectura de Red Mejorada

Apache2, envío de diffs binarios entre cliente y servidor, datos comprimidos con mod_deflate.

Navegando por el repositorio

Puede apuntar su navegador a la URL del repositorio y navegar por sus contenidos sin tener un cliente de Subversion. Esto da acceso a sus datos a un mayor círculo de usuarios.

Autenticación

Puede utilizar cualquier mecanismo de autenticación que Apache soporte, incluyendo SSPI y LDAP, Active Directory (Directorio Activo).

Seguridad

Dado que Apache2 es muy estable y seguro, automáticamente obtendrá la misma seguridad para su repositorio. Esto incluye la encriptación SSL.

MIME types y detección automática de ficheros binarios.

Permite operar directamente sobre el repositorio sin copia local.

Permite backups en caliente.

Por otra parte subversión incluye un cliente remoto (svn), y un servidor (svnserve), y muchas utilidades. Pero en este documento solo se hace alusión al uso de apache debido a que permite más posibilidades que svnserve. Entrando en materia, lo primero que debemos hacer es instalar **apache2, subversion, subversion-tools**. Luego de instalar estos paquetes podemos instalar **libapache2-svn**, para que apache transforme las operaciones http de los clientes. Este paquete instala el modulo **mod_dav_svn**, útil para lo antedicho.

Crear un repositorio de subversión

Primero se crea una carpeta donde radicará el repositorio:

```
mkdir -p /servidor/svn/UNICORNIOS
```

Se crea el repositorio:

```
svnadmin create /servidor/svn/UNICORNIOS
```

Permisos al servidor Web:

```
chown R wwwdata:wwwdata /servidor/svn/UNICORNIOS
```

Estas operaciones se realizan como root de la máquina donde usted esté creando el repositorio.



Pasos en la configuración principal

Posteriormente configuraremos el fichero del apache donde se añadirá un virtualhost para entrar al subversión vía apache, y dentro de este editar las líneas necesarias para configurar el `mod_dav_svn`. El fichero está en `etc/apache2/sitesavailable/default`. Se agregan estas líneas al final del fichero:

```
<VirtualHost 10.33.20.213>
ServerAdmin Webmaster@localhost
ServerName 10.33.20.213
<Location /svn>
DAV svn
# Establecer el camino al repositorio subversion
SVNPath /servidor/svn/UNICORNIOS
AuthType Basic
AuthName "Subversion Repository"
AuthUserFile /etc/subversion/passwd
# To enable authorization via mod_authz_svn
AuthzSVNAccessFile /etc/apache2/dav_svn.authz
PerlAuthenHandler Apache2::AuthenMSAD
PerlSetVar MSADDomain uci.cu
PerlSetVar MSADServer 10.0.0.3
Require valid-user
</Location>
</VirtualHost>
```

La parte inicial del bloque nos dice que nombre tendrá en la Web el sitio del repositorio, es decir, por cual dirección http nos comunicaremos al subversion. En este caso sería: <http://10.33.20.213> y esta línea que viene a continuación es donde empieza la configuración del `mod_dav_svn`.

<Location /svn> (significa que los repositorios de Subversion están disponibles en la URL <http://10.33.20.213/svn/>)

DAV svn (le dice a Apache qué módulo será responsable de servir esa URL - en este caso, el módulo de Subversion.)

SVNPath /servidor/svn/UNICORNIOS (le dice a Subversion que busque repositorios bajo /servidor/svn/UNICORNIOS)

AuthType Basic (se utiliza para activar la autenticación básica, es decir, Usuario/contraseña)

AuthName "Subversion Repository" (se utiliza cuando le aparezca un diálogo de autenticación al usuario como información para decirle para qué se necesita su autenticación)

AuthUserFile /etc/subversion/passwd (especifica qué fichero de contraseñas se utiliza para la autenticación)

AuthzSVNAccessFile /etc/apache2/dav_svn.authz lugar del fichero de acceso para las rutas dentro del repositorio de Subversion

Require valid-user (especifica que sólo los usuarios que hayan introducido un par usuario/contraseña válido podrán acceder a la URL)

Autenticacion por Active Directory (Directorio Activo)

Las otras líneas del bloque:

PerlAuthenHandler Apache2::AuthenMSAD

PerlSetVar MSADDomain uci.cu

PerlSetVar MSADServer 10.0.0.3

Este es el método en que nosotros nos autenticamos en nuestra universidad. Pero para lograrlo es necesario que se instale un módulo de perl que nos permite configurar el apache para esto. Cuando haga esto ponga las líneas indicadas arriba como mismo esta en la configuración anterior, y reinicie el apache:



```
/etc/init.d/apache2/restart
```

Luego acceda a <http://10.33.20.213/svn> y podrá corroborar los resultados.



Perl en sólo una línea

Jorge Luis Rodríguez Carpio
jrodriguez@estudiantes.uci.cu
Proyecto FILPACON

Perl es un lenguaje de programación diseñado por Larry Wall y creado en 1987. Toma características de C, del lenguaje interpretado shell (sh), AWK, sed, Lisp y, en grado inferior, de muchos otros lenguajes de programación. Estructuralmente, Perl está basado en un estilo de bloques como los de C o AWK, y fue ampliamente adoptado por su destreza en el procesamiento de texto y no tener ninguna de las limitaciones de los otros lenguajes de script.

Disponibilidad

Está especialmente extendido en Unix y en sistemas similares (como en Linux, FreeBSD y Mac OS-X) pero también se le puede encontrar en sistemas Windows. Perl está licenciado bajo la Licencia Artística y la GNU/GPL. Ha sido adaptado a más de un centenar de plataformas diferentes. Una adaptación especial, MacPerl, está disponible para Mac OS Classic. Perl está instalado por defecto en las distribuciones más populares de Linux como Gentoo, Debian y SuSE.

Perl puede ser compilado desde el código fuente para Windows; sin embargo, muchas instalaciones Windows no disponen de un compilador de C, por lo que los usuarios de Windows normalmente instalan una distribución binaria, como la de ActivePerl, IndigoPerl y otras más.

Utilidad del lenguaje

Perl es utilizado en muchas tareas de administración de sistemas POSIX porque, entre otras ventajas, dispone de un grupo de parámetros de llamada del propio intérprete que hacen que éste haga cosas increíbles en tan sólo una línea. La intención de este pequeño artículo es demostrar ese poder que tiene Perl para hacer cualquier cosa de forma sencilla y rápida.

Facilidades que ofrece Perl

Ejecutar expresiones Perl: -e

Este parámetro, que puede repetirse varias veces, permite ejecutar expresiones de Perl en sólo una línea. No se buscarán nombres de programa en los restantes argumentos y se añade un salto de línea al final de cada expresión.

Ejemplo: `$ perl -e 'print "Hola\n"' -e 'print "y adiós \n"`

Lectura y proceso de la entrada: -n

Envuelve todas las expresiones de código alrededor de lo siguiente:

```
LINE:
while (<>) {
... # aquí va nuestro código
}
```

quedando de la siguiente manera:

```
perl -n -e "\$_ =~ s/juan/pepe/i; print;" tmp.txt
```

Es decir, lee la entrada estándar (en este caso el contenido dentro del fichero tmp.txt), línea a línea, y ejecuta el código indicado dentro del bucle. En este ejemplo donde quiera que aparezca la palabra “juan” dentro del texto tmp.txt se cambia “pepe”. Para comprender mejor el funcionamiento de “s/juan/pepe/i” consultar la ayuda de Perl para expresiones regulares (perlre).



Lectura y proceso de la entrada con copia a la salida estándar: -p

Envuelve todas las expresiones de código alrededor de lo siguiente:

```
LINE:
while(<>){
... # aquí va nuestro código
} continue {
print;
}
```

Entonces el ejemplo anterior puede quedar resumido de la siguiente manera:

```
perl -n -p -e "\$_ =~ s/carpio/pepe/i;" tmp.txt
```

Donde el "print" se sustituye por el parámetro "-p". y obtenemos el resultado esperado. Lo que provoca que, tras ejecutar nuestro código, en cada pasada del bucle, envía a la salida estándar del proceso el contenido de la variable \$_.

Partir la entrada en campos: -a

Utilizado con -p ó -n habilita un preproceso de cada línea de entrada, efectuando una división por espacios mediante la función split sobre una lista @F. El separador puede cambiarse utilizando el parámetro -F.

```
LINE:
while (<>) {
@F = split(' ');
... # aquí va nuestro código;
}
```

Ejemplo: `perl -ane 'print pop(@F), "\n";' tmp.txt`

Separa la entrada estándar creando una lista para cada línea del fichero tmp.txt e imprimiendo sólo la última posición del mismo.

Ejemplo usando el parámetro -F: `perl -ane 'print shift(@F), "\n";' F:' /etc/passwd`

El ejemplo anterior imprime todos los nombres de los usuarios existentes en la PC, leídos desde el archivo "/etc/passwd". El funcionamiento radica en que se cambio el carácter para dividir por las líneas (por defecto un espacio " ") por el el doble punto ":"

Conclusiones

La solución que propone Perl con el lema 'Hay más de una manera de hacerlo' ("There's more than one way to do it."), es ofrecerle al programador libertad en su forma de expresarse, ¿Acaso todos hablamos el mismo idioma?, ¿Dónde queda la imaginación?, ¿Pasó a ser un lujo?. Los novatos en programación se sorprenden con la cantidad de cosas disponibles y lo catalogan de lenguaje muy complicado y no pueden comenzar a apreciar la flexibilidad que encuentran si solamente han programado en lenguajes tradicionales imperativos del nivel que sea, enmarcados en "entornos de programación" (Too ugly for words) y con "desarrollo rápido de aplicaciones" (vim-perl y cerebro, eso es todo lo que hace falta).



Optimizar el arranque en sistemas GNU/Linux

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

Este artículo le presenta al lector una solución para optimizar el arranque del sistema operativo, está basado en la Distribución Debian GNU/Linux, aunque se puede utilizar para Ubuntu GNU/Linux teniendo en cuenta que en ese caso, no existe el archivo de configuración `/etc/inittab`, sino que el contenido del fichero antes mencionado se encuentra compartido en varios ficheros de configuración dentro del directorio `/etc/event.d/`. Se recomienda como referencia para todos los sistemas GNU/Linux, pues la filosofía no cambia para ninguna distro, solo la organización en el sistema de ficheros.

Introducción

Todos deseamos un sistema que arranque "super rápido", pero al quitar los permisos de ejecución a los scripts de `/etc/init.d/`, como aconsejan algunas guías, este sigue demorando, teniendo en cuenta, además, que de esta forma:

- Se provocan errores en el inicio del Sistema.
- Se viola la cadena lógica establecida en las normas de Debian (Debian-Policy) concerniente a la configuración de los niveles de ejecución.

Comenzar correctamente

Para los sistemas GNU/Linux el proceso inicial, padre de todos los procesos, es `init`. Ejecutando el comando `pstree` se puede ver gráficamente la idea planteada. El fichero `/etc/inittab` y las carpetas `/etc/rc*.d` (donde * puede tomar los valores 0-6 o S) serán el centro de "ataque" para la nueva configuración.

El Sistema en su inicio tiene 3 partes:

- La carga de la imagen del kernel y módulos
- El `sysvinit` (scripts de inicialización y configuración)
- Los `runlevels`

Carga de la imagen del kernel y módulos

La carga del kernel le toma al Sistema de 6 a 7 segundos, tiempo que puede reducirse compilando este, lo cual nos permite quitar soporte para el hardware y protocolos que no utilizamos. El objetivo es lograr una imagen pequeña y modular que se cargue más rápido. Luego de esto tendremos una carga de 4 a 5 segundos; llevarla a 3 es un reto bastante fuerte.

Sysvinit (scripts de inicialización y configuración)

En este punto es donde tiene lugar la mayor cantidad de procesos, esta etapa determina que la PC esté lista en los próximos 7 a 20 segundos para comenzar a trabajar.

Runlevels

En los `runlevels` se adicionan procesos al arranque (ej: `apache2`, `vsftpd`, `zope2.9`, `gdm`, etc); algunos de estos se pueden configurar para que comiencen cuando el usuario inicie la sesión y no en el arranque. Con los primeros 2 pasos bien implementados se tiene una base de 8 a 10 segundos. Los procesos que se disparen en el `runlevel` largarán este tiempo para el inicio (ej: con `ssh`, `gdm`, `apache`, `vsftp` y `zope2.9`, se pueden lograr 15 segundos, solo con `ssh` 12 segundos).

El Sistema tiene 7 `runlevels`:

runlevel 0: Parada del sistema(no usar para la carga del sistema)

runlevel 1: monousuario

runlevel 2: multiusuario sin red

runlevel 3: multiusuario con red



runlevel 4: no se usa: multiusuario

runlevel 5: multiusuario con interfaz gráfica

runlevel 6: reinicio(no usar para la carga del sistema)

En el fichero inittab se define cuál será el runlevel que cargará el Sistema en el arranque y quien lleva el control de los procesos que se ejecutaran en el sysvinit.

Dentro del fichero inittab, la línea:

id:#:initdefault:

donde # es el número del runlevel que se va a cargar por defecto en el arranque del sistema.

La siguiente línea nos indica quién lleva el control de los scripts de inicialización y configuración del sistema:

si::sysinit:/etc/init.d/rcS

Relación del inittab con las carpetas /etc/rc*.d/

Cada runlevel tiene un lugar que define los procesos que se ejecutan, estas carpetas son: /etc/rc#.d/ donde # representa un número, que coincide con el número de runlevel, es decir:

/etc/rc1.d/ es del runlevel 1

/etc/rc2.d/ es del runlevel 2

/etc/rc3.d/ es del runlevel 3

/etc/rc4.d/ es del runlevel 4

/etc/rc5.d/ es del runlevel 5

/etc/rc6.d/ es del runlevel 6

Queda por definir el directorio /etc/rcS.d/ Este directorio es el objetivo del script /etc/init.d/rcS que controla nuestro sysvinit y en el están todos los procesos que se ejecutarán.

¿Cómo quitar los procesos?

Los enlaces simbólicos de los procesos que no se utilizan de la carpeta /etc/rcS.d se quitan y se renombran los de las carpetas /etc/rc#.d/ (donde # es el runlevel que dejamos por defecto).

Renombrar los enlaces simbólicos

En los runlevels no sólo se inician procesos sino también se matan procesos, luego el tema principal es: ¿cómo hacer la diferencia? Los enlaces simbólicos se nombran de acuerdo a las dos acciones que se pueden realizar y se sigue el siguiente formato:

acción|orden_ejecución|nombre_proceso

Para definir las posibles acciones a realizar tenemos:

S: acción Start

K: acción Kill

El orden de ejecución es un número decimal que define el orden de los procesos, eliminando de esta forma el problema que pudiera existir en virtud de la dependencia de los procesos, garantizando el correcto orden de ejecución.

Ejemplo:

Un proceso iniciado en el runlevel por defecto y que no se utiliza, sólo debemos renombrarlo, ej: Se tiene:

S25apache2

Se cambia a: K25apache2



Herramientas

Existen algunas herramientas que ayudan a quitar los servicios al nivel de los runlevels, básicamente lo que hacen es renombrar los ficheros que hacen referencia a los servicios que no queremos, estás son:

Interfaz Ncurses: rcconf

Interfaz Gtk: bum



LinuXierra, primera comunidad de Software Libre en las montañas cubanas

Ing. Abel Meneses Abad, Evelio Clavel Rosales, Daniel Marino Mirayes Tasset
abelma@uci.cu, evelio@LinuXierra.org, dmmirayes@estudiantes.uci.cu
Comunidad LinuXierra

El 28 de Abril del 2007 presenciamos con orgullo cómo, por primera vez, cuatro ciudades cubanas se inscribieron en el Festival Latinoamericano de Software Libre. Cualquiera que conozca la República de Cuba, hubiera experimentado la sensación de la duda al ver inscrita entre estas cuatro, a la “ciudad” de Buey Arriba¹⁷.

A partir de esa fecha hemos conversado y difundido la idea de cuán importante puede resultar para una comunidad de montaña la existencia del software libre, y más aun cuánto de sabio y posible puede tener el desarrollo comunitario usando software libre.

Buey Arriba

Es un municipio montañoso situado en la provincia Granma de la zona Oriental de Cuba¹⁸. Con más de 32 000 habitantes¹⁹, alcanza una de las tasas de mortalidad más bajas del país. Cuenta con un policlínica docente, un hospital y varios consultorios médicos. Tiene escuelas primarias, secundarias, una sede pedagógica, una deportiva, una de ciencias médicas y una universitaria. Sobresale la existencia de dos Joven Clubs de Computación y Electrónica que brindan servicios a la comunidad. Posee un canal local de TV: MinasVisión, además de contar con la multipremiada productora de documentales TV Serrana. El cultivo fundamental es el café que se comercializa mundialmente.

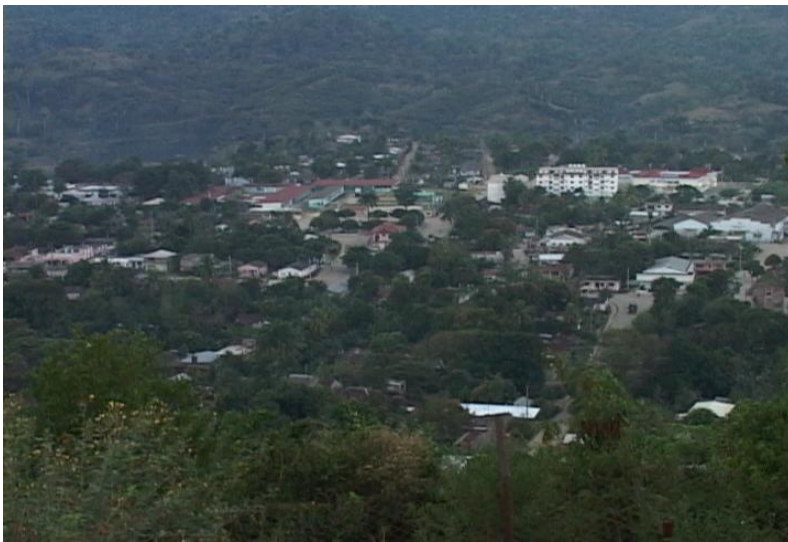


Figura 14: Vistas aéreas del municipio Buey Arriba

¹⁷ Evelio Clavel Rosales, Buey Arriba en FLISOL 2007, <http://softwarelibre.uci.cu/modules/news/article.php?storyid=2502>

¹⁸ Anónimo, Buey Arriba, 20 abr 2007, http://es.wikipedia.org/wiki/Buey_Arriba

¹⁹ Situación de Salud en Granma. Indicadores Básicos 2002, <http://www.sld.cu/galerias/pdf/sitios/dne/granma2002.pdf>



¿Qué es LinuXierra?

Precisamente en uno de los dos Joven Clubs²⁰ que el país a construido en este municipio de montaña, aproximadamente desde el año 2001 se promueve por un grupo de jóvenes el uso de Software Libre. LinuXierra es el Grupo de Usuarios de Software Libre de Buey Arriba. Se fundó el 28 de Abril de 2007 gracias al entusiasmo de algunas personas interesadas en compartir sus experiencias en el uso del Software Libre, específicamente en GNU/Linux, así como para apoyar su difusión²¹.

Un poco de historia...

Como casi todas las comunidades, la existencia de estas casi siempre tributa a la aparición de líderes. No precisamente “intelectuales” preparados, sino a personas una capacidad para perseverar, una férrea voluntad y una visión de futuro acompañada de un sueño de humanidad, les confieren esta condición. A este tópico habrá que dedicarle innumerables horas con posterioridad.

La comunidad – o las actividades de Software Libre – en Buey Arriba tienen sus inicios en un 486, que usaran Yandys Cervantes Rodríguez y Daniel Marino Mirayes con FloPPyX, una distro que usaron en aquella PC sin disco duro. Por ese entonces el primero era estudiante de la Universidad de Oriente, y Daniel estudiante de Preuniversitario (más tarde comenzaría a estudiar en la UCI). Yandys, debió conocer la comunidad de la UO de la cual integraron luego como profesores Sergio, Alain y otros importantes fundadores de la comunidad de SWL de la UCI. Conoció las distros que se usaban en PCs de pocas prestaciones – abundantes en la zona oriental de Cuba – y con las cuales se hacían numerosos trabajos. Se trasladó al Joven Club II de Buey Arriba donde comenzaron a hacer pruebas.

Evelio Clavel Rosales comenzó en noveno grado (2001) con Knoppix; y un par de años más tarde comenzaría sus investigaciones sobre Ubuntu 4.10. Trabajó con Mandrake, Red Hat, entre otros.

Los años que anteceden al 2006...

Yandys continuó sus trabajos en la Universidad de Granma, y mientras Daniel comenzaba a estudiar en la UCI y se integraba rápidamente a la comunidad de SWL allí, Evelio continuó sus estudios en la Vocacional de Ciencias Exactas. La práctica en cada uno de los niveles de enseñanza puede resultar algo traumático, pues en Cuba – gracias al no pago de licencias por el bloqueo – la inserción del SWL en la educación transita por caminos aún lentos.

La Universidad de Granma, al igual que el resto en el país, tiene por evaluación al menos el 70% de sus servidores con sistemas operativos libres. Esto no es mucho problema si partimos de la preferencia internacional por los servidores Web Apache y Bases de Datos, MySQL y PostgreSQL. El resto de las condiciones son en el orden de los 90% software privativo. Salvo algunas iniciativas independientes el resto usa Windows y no conocen otra cosa que software privativo.

En el preuniversitario es similar, aunque algunos de los linuxeros de Buey Arriba utilizaron ya en el pre: Knoppix y Ubuntu. Vale destacar sin embargo que toda la enseñanza preuniversitaria tiene laboratorios y un numeroso conjunto de softwares para la enseñanza. Siempre de alguna manera tener a mano un Knoppix permite burlar algunas de las vigilancias técnicas, y lo prohibido gusta mucho a los adolescentes. De los bueyarribenses el más afortunado ha sido sin dudas Daniel que dentro de la Universidad de las Ciencias Informáticas encontró un ambiente – en sus inicios algo hostil – pero que ya en su 3er año de fundada se consiguió una facultad completa para el Software Libre, y al término de estos primeros 5 años comienza a dar saltos agigantados hacia las plataformas libres en el ámbito productivo y docente.

²⁰ 2007, <http://www.jovenclub.cu/>

²¹ ¿Qué es el LinuXierra?, 4 jun 2007, <http://www.LinuXierra.org/>



En disímiles condiciones se conservó en los años siguientes antes del 2006 y posterior al 2001 cada cual en su escuela el amor por el Software Libre con todos sus preceptos ideológicos y tecnológicos. Fueron años de aprendizaje y de consolidación en el trabajo en casa.

El IPVCE “Silberto Álvarez Arocha”

Evelio continuó sus estudios en la vocacional, allí logró que varios compañeros se sumaran a sus iniciativas. Algunos por amor al Software Libre y otros para jugar con lo prohibido. Se incorporó también al equipo de programación, con conocimientos muy básicos de “Basic” logró mantenerse en el equipo y avanzar en sus estudios. Evelio había aprendido este lenguaje en su 7mo grado, cuando Yandys siendo instructor del Joven Club le había enseñado a trabajar con las MSX-Basic que quedaban en su secundaria.

¿Qué tiene que ver los concursos de informática en el preuniversitario cubano con el Free Software?

“mucho, pues usábamos FreePascal y DJGPP. En 12mo teníamos un LiveCD que le regalaron al entrenador en una IOI, basado en Knoppix, empaquetada por la gente del Code Up 2006 y la IOI, programé en los IDEs que traía para Pascal y C/C++ ...”²²

Xierra Team

Antes de la conformación de LinuXierra, Yandys, Albio Ernesto Figueredo (entonces técnico de educación) y Daniel tenían la idea de hacer un grupo de usuarios al que pensaban denominar Xierra Team. Pero la falta de recursos, la sobrecarga de tareas de los líderes (universidad – trabajo) hicieron fracasar la primera idea.

Nacimiento de LinuXierra (Etapas Final)

Después de las improvisadas instalaciones y su incipiente incorporación en la vocacional, Evelio pasó su servicio militar donde encontró de manera casual “Nova Lnx” (así se llamó en una época a Nova). Pudo terminar felizmente su trabajo, y ya tenía conocimientos de ellas pues Daniel le había comentado su existencia.

Se incorporó luego al Joven Club II de Buey Arriba donde retomó la idea de refundar el grupo de usuarios de GNU/Linux de Buey Arriba, dando origen a la comunidad que nos absorbe en este trabajo: LinuXierra. Allí con la ayuda de Olexis Barrero Castillo trabajador del Joven Club, se impartió el primer curso de GNU/Linux usando la distribución Mandriva PowerPack 2007 (esfuerzo conjunto Francia – Brasil). Febrero 2006, Aula 2 del Joven Club. Las circunstancias hicieron que fuera Mandriva la ideal, pues fue la que pudieron conseguir en Educación (con Albio, un técnico amigo) y utilizarían el entorno KDE muy parecido al de XP.

Evelio incluyó poco después a Buey Arriba en el FLISOL (circunstancia que diera lugar a este fructuoso intercambio y a la promoción de LinuXierra). Y posteriormente se informó a través de los Joven Club que se haría un Festival de Instalación. Y se efectuó la idea arriesgada con el apoyo de las autoridades del municipio. Esta fue la fundación oficial de LinuXierra. Como parte de las actividades la página del Joven Club la cambiaron a Joomla, y escribieron a otros Joven Club invitándolos a hacerlo. Jiguaní y Manzanillo se sumaron a la promoción de SWL usando Joomla. Posteriormente se han comenzado a estudiar como parte de estos esfuerzos otros CMS, aplicándolos a los Joven Club.

La cuarta actividad del grupo – para mi la más interesante de todas – fue la de colocar a los niños del municipio a probar los juegos de GNU/Linux. Logrando incluso mejor comprensión de algunos como Gcompris, respecto a otros de Windows. Este es uno de los temas en los que habrá que trabajar muy fuerte.

²² Tomado de Evelio Clavel Rosales, email enviado a Abel Meneses Abad, 21/05/2007



*Figura 16: Comunidad LinuXierra,
abril 28, FLISol 2007*



*Figura 15:
Guajitux, mascota
de LinuXierra*

Conclusiones

LinuXierra es la primera comunidad rural de Software Libre, y para colmo de sorpresas de montaña de la que tenemos conocimientos en Cuba. Su existencia nos deja claridad del esfuerzo y el sacrificio del que son capaces los seres humanos aun en los lugares de menor ancho de banda imaginable. Queda pues la osadía de llegar hasta allá y apoyar a este grupo con todo lo que tengamos a mano. De seguro otras comunidades comprenderán la importancia del Software Libre, y la independencia tecnológica que este que podría generar para ellas, además de una vida intelectual más austera en función del ser humano, algo imposible en software privativo.



Diseñar con Blender en GNU/Linux

Yunier Vega Rodríguez
yvrodriguez@estudiantes.uci.cu
Proyecto UNICORNIOS

“Lo más importante no es la herramienta, sino lo que deseamos crear. Por ese motivo, estas deben servirnos y adaptarse a nuestros deseos ... no al revés.”

Diana Thorneycroft

Cuando se habla de los sistemas GNU/Linux, la gente por lo general asocia el concepto a una consola que sólo los informáticos entienden, despojada de un ambiente interactivo asequible y una usabilidad diversa. Nada más lejos de la realidad, pues en GNU/Linux todo es posible, incluso “Diseñar”, y “Diseñar con Calidad”. Herramientas hay muchas, y todas destacan por su eficiencia en el tratamiento de la imagen y la creación de objetos y animaciones en los espacios geométricos conocidos. A una de ellas específicamente he querido dedicar este artículo, debido en gran medida al desconocimiento que existe acerca de sus potencialidades.

Blender, un proyecto multiplataforma que tuvo sus orígenes en el Software Libre, es por así decirlo, el *Rey del Diseño y la Animación 3D en GNU/Linux*. Caracterizado por ser una aplicación realmente ligera con un ambiente de fácil manejo para el neófito respecto a otros software de su misma especie, la concepción de su estructura es el resultado de la implementación de las mejores prácticas para el diseño.

Un poco de historia

Cuando en 1988 Ton Roosendaal co-fundó en Holanda el estudio de animación Neo Geo, nunca imaginó que años más tarde (1995) estaría escribiendo el código de lo que sería Blender, mucho menos que habría de crear en la primavera de 2002 Blender Foundation, una organización no lucrativa encargada de promocionar *Blender Publisher* en todo el mundo, a fin de que su código fuera liberado a la comunidad de usuarios, luego de que los inversores de NaN desestimaran la continuidad de su desarrollo, lo cual ocurrió el domingo 13 de octubre de 2002 bajo los términos de la Licencia Pública General (GPL) de la Free Software Foundation.

¿Qué hace de Blender un software de calidad?

Blender es un paquete integrado de herramientas, para la creación de un amplio rango de contenidos 3D, ofreciendo total funcionalidad para el modelado, renderizado, animación, post-producción, creación y reproducción de contenido interactivo con los singulares beneficios de ser multiplataforma. Destinado a artistas y profesionales, Blender puede ser usado para crear visualizaciones 3D, tanto estáticas como de video de alta calidad, mientras que la incorporación de un motor para tiempo real, a más del poderoso motor de juegos de la suite 3D, permite la creación de contenido interactivo que puede ser reproducido independientemente.



Figura 17: Ejemplos de diseño en Blender

En la actualidad Blender va por la versión 2.44, con la inclusión de algunas mejoras y utilidades recientemente que lo sitúan sin lugar a dudas en un lugar privilegiado para los que como nosotros, no contamos con el capital suficiente que nos permita adquirir software de la magnitud propietaria de 3DMaxStudio, cuyas funcionalidades, muchas veces tienden a ser sobrepasadas por esta herramienta.

Características de la interfaz de Blender

Si es nuevo en Blender, debería asimilar como trabajar con la interfaz de usuario antes de ponerse a modelar. Los conceptos detrás de la interfaz de Blender hacen que esta no sea muy estándar, sino que sea diferente de los otros paquetes 3D. Especialmente los usuarios de Windows necesitarán coger el truco a la manera en la que Blender maneja los controles, tales como botones o movimientos de ratón. Pero esta diferencia es de hecho la gran fuerza de Blender: una vez comprenda la manera en la que funciona este, encontrará que puede hacer su trabajo extraordinariamente rápido y de manera muy productiva.

La figura 11 muestra la vista de pantalla que debe obtenerse al haber iniciado Blender. Por defecto está separada en tres ventanas: el menú principal en la parte alta, la vista 3D grande y la Ventana de Botones abajo. La mayoría de las ventanas tiene un encabezado que puede ubicarse en la parte alta o en la parte baja del área de una ventana.

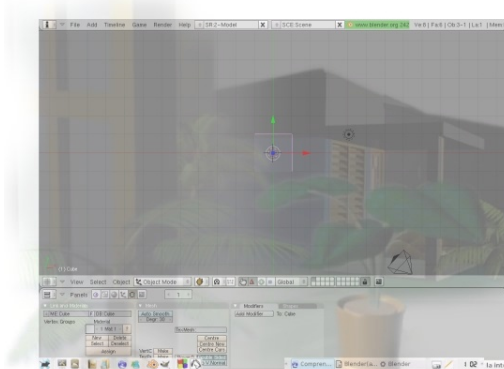


Figura 18: Interfaz de inicio de Blender

Es posible personalizar el sistema de ventanas de Blender para satisfacer las necesidades y gustos del usuario. El marco de cada ventana puede contener diferentes tipos y conjuntos de datos, dependiendo de lo



que se está haciendo. Estos pueden incluir modelos 3D, animación, materiales de la superficie, scripts Python, etc.

Blender en la UCI

Como indican las estadísticas, existe en la Universidad muy poca experiencia en el uso de Blender, y en general en el empleo de herramientas libres para el diseño, a lo que se une el hecho en sí de que la mayoría de la Comunidad Universitaria sea asidua de las herramientas propietarias, quizás por desconocimiento o temor infundado, a enfrentarse a herramientas que temen sea de menor potencia y prestaciones. Con el texto de este artículo, a más de los ejemplos que se exponen en él, se ha querido demostrar la inconsistencia del criterio de algunos para los que el Software Libre continúa siendo la cara oculta de una consola envuelta en un lenguaje ilegible, mostrándonos además que en verdad el único tabú para crear somos nosotros.



Una vía para la enseñanza de GNU/Linux en los Jóvenes Club

Msc. José Candelario Balmaseda, DrC. Hernán Fera Avila, Lic. Alberto Méndez Pérez
cande02034@itu.jovenclub.cu, hernan@itu.rimed.cu, mdz@villazul.itu.sld.cu

PROPUESTA DE PROGRAMAS PARA LA ORGANIZACIÓN CURRICULAR DE LA ENSEÑANZA-APRENDIZAJE DEL SISTEMA OPERATIVO GNU/LINUX EN LOS JÓVENES CLUB DE COMPUTACIÓN Y ELECTRÓNICA.

Con el desarrollo y el uso masivo de las TIC en Cuba ha comenzado una nueva etapa en la educación; en ella la computación se constituye en uno de sus recursos tecnológicos de mayor trascendencia. En este sentido los Jóvenes Club de Computación y Electrónica desempeñan un papel esencial, ya que por sus instalaciones pueden y deben pasar todos los estratos de la población.

Este movimiento cuenta con una vasta red de instituciones a todo lo largo y ancho de nuestro país, que tiene como misión "Proporcionar una cultura informática a la comunidad con prioridad hacia niños y jóvenes, jugando un papel activo, creativo y de formación de valores en el proceso de informatización de la sociedad cubana". El curso de mayor demanda, entre los que se ofertan en los Jóvenes Club, es el de Operador de Microcomputadora, que inicia a los estudiantes en el Sistema Operativo Microsoft Windows. Sin embargo, en estos momentos es una necesidad y política de nuestro estado "... la migración progresiva de las computadoras instaladas en los organismos de la Administración Central del Estado hacia el software libre, sobre la base del sistema operativo Linux, eliminando así la presencia casi exclusiva del Windows en las máquinas". Se requiere aclarar, en relación con esta cita, que el nombre correcto es GNU/Linux, ya que Linux se refiere, en lo fundamental, al Kernel o núcleo del sistema.

¿Por qué migrar hacia el software libre?

Entre los software, los más importantes son los relacionados con los sistemas operativos. De ellos, durante años se ha venido enseñando solo uno: Microsoft Windows, el que hoy día, en nuestro país y otras partes del mundo, está instalado en la inmensa mayoría de las computadoras personales. Sin embargo, es un sistema operativo que es del tipo propietario, es decir, que para usarlo debemos comprarlo y regístralo por la licencia copyright que ampara este tipo de programas.

Según las consultas bibliográficas realizadas por el autor, el movimiento de software libre presenta aspectos positivos tales como, el no ser monopolístico; el oponerse a las patentes; el cuestionar la propiedad privada del conocimiento; el no tener un centro rector o país hegemónico; el basarse en un método de trabajo cooperativo; y lo más importante, trascender el modo de producción en el que fue engendrado, cuya contradicción principal se daba entre la producción social y la apropiación privada de sus resultados. En él, tanto la producción como la apropiación del software es social.

Las bondades antes descritas, para nuestro bloqueado país, son esenciales. Continuamente se nos niegan patentes y tecnologías, e incluso publicaciones, participación en congresos y ventas de productos soportados en tecnología Microsoft Windows. A diferencia de esta férrea posición, Cuba es paladín de la educación y transmisión gratuita de conocimientos. Afortunadamente existe una opción: usar un software del tipo libre, que podamos adaptarlo a cualquier necesidad y distribuirlo por todo el país sin ningún problema legal. Según Roger Peña, en una entrevista concedida a La ventana, Portal Informativo de La Casa de las Américas en el artículo Entrevistas: Enviado el viernes, 7 de Mayo del 2004 "... La mayoría de la gente cree que Windows es más fácil de usar que Linux, que este es solo para entendidos, pero no es exactamente así. Lo que sucede es que existe muy poco conocimiento de Linux y por tanto falta personal humano que entrene a quienes quieren adentrarse en este mundo.

Esto se aprecia en los diferentes niveles de enseñanza de nuestro país, por cuanto, como barreras subjetivas se subutiliza en los pocos centros que lo tienen instalado actualmente y se subestima por la mayoría de profesionales que se inician en su conocimiento.



En GNU/Linux hay una potencialidad que debe ser explorada y explotada. Como sistema operativo es totalmente funcional y la distribución del mismo, que se usa en tres de los cinco Jóvenes Club de Computación y Electrónica del municipio Puerto Padre, la Tinux, basaba en una remasterización de OpenSuSE, es compatible con las aplicaciones del Microsoft Office.

La tesis en opción al Título Académico de Master en Nuevas Tecnologías para la Educación “PROPUESTA DE PROGRAMAS PARA LA ORGANIZACIÓN CURRICULAR DE LA ENSEÑANZA-APRENDIZAJE DEL SISTEMA OPERATIVO GNU/LINUX EN LOS JÓVENES CLUB DE COMPUTACIÓN Y ELECTRÓNICA” sugiere una vía para la generalización de la enseñanza de este sistema operativo. Primeramente se partió de hacer una nueva distro (remasterización) que fuera amigable para el usuario, el Licenciado Alberto Méndez Pérez, quien es el padre, nos aclara que partió de la necesidad de una distribución fácil para su uso. Adaptable para la migración de Windows a Linux. Amigable al usuario. Que incluya los paquetes básicos de Oficina. Pocos CD de instalación para su generalización.

¿Por qué OpenSuSE?

Reúne todas las condiciones anteriormente mencionadas. Se basa en software libre, no comercial. Su centro de administración, Yast, hace más fácil la instalación y administración del sistema, característica que lo diferencia de las demás distribuciones y lo hace menos hostil al administrador del sistema y al usuario en sí. Esa propuesta consta de tres programas, uno de operador de microcomputadora sobre GNU/Linux, otro de usuario avanzado de Cálculo, el tabulador electrónico u hoja de cálculo, similar y que no le cede en nada, al Microsoft Excel y un programa para iniciar al estudiante en la programación en C++.

¿Por qué estos tres programas?

Pensemos que mañana despertáramos con la noticia de que no podemos usar Microsoft Windows, ¿Cómo preparar en breve tiempo a los usuarios que se inicien en GNU/Linux? Un curso de operador de microcomputadora desde cero, para personas que nunca han oído hablar de informática es necesario, en el se incluyen elementos de sistema operativo, de ofimática, (procesador de texto, hojas de cálculo, presentaciones y base de datos) correo electrónico y redes.

El curso para usuarios avanzados de hoja de cálculo es fundamental, hasta que se construyan sobre software libre los principales ERP (Enterprise Resource Planning o Planificación de Recursos Empresariales) las empresas deben resolver inmediatamente numerosos problemas que pueden solucionarse de esta forma. La inmensa mayoría del código fuente que acompaña al software libre es hecha en C++, de ahí que un curso que inicie a los estudiantes en este lenguaje de programación sería vital para su comprensión y desarrollo.

Estos cursos están confeccionados basándose en 64 horas clases, con 32 encuentros (dos semanales) de dos horas de duración. Se comenzaron a aplicar, el de operador de microcomputadoras, en octubre del 2005 y hasta la fecha (contando el que se está desarrollando actualmente) se han hecho cuatro, el de cálculo se aplicó de octubre del 2006 a enero del 2007 y el de programación den C++ se aplicado dos veces, de octubre del 2006 a enero del 2007 y de marzo a Junio del presente año.

La aceptación por parte de los estudiantes ha sido muy buena y ha existido un buen aprovechamiento de las clases impartidas. Creemos firmemente que debemos estudiar cada día más para demostrarles a los incrédulos que existe otra filosofía que debemos aplicar, debemos buscar cada día nuevos caminos para demostrarle a los fanáticos del Microsoft Windows, esos que no piensan, esos que no tienen paz, esos que repiten frases de otros que Un mundo mejor es posible.

Como bien dijera el venezolano José Luis Regalado. "Donde hay ignorancia hay fanatismo, donde hay fanatismo no hay tolerancia, donde no hay tolerancia no hay paz" Conozcamos a GNU/Linux en particular y al Software Libre en general para eliminar la ignorancia, para dejar de ser fanático y alcanzar la paz.



PHP5 vs. PHP4

Rolando Herrera Martínez
rherrera@estudiantes.uci.cu
Universidad de las Ciencias Informáticas

Después de un tiempo de interacción con este magnífico lenguaje de programación que es PHP, me he dado a la tarea de realizar un artículo sobre PHP5 vs PHP4, aclaro desde un inicio que no soy experto de PHP4, pero espero mostrarles algunas cosas que nos hacen pensar: “¿Es PHP4 la mejor elección para desarrollar aplicaciones Web?”.

Es importante decir a los lectores que se inician en este lenguaje, o que sólo están leyendo por pura casualidad e interés, que PHP surge como un lenguaje personal del desarrollador Rasmus Lerdorf para hacer sus páginas Web más fáciles de crear. PHP era un lenguaje de script con mucha rapidez, pero insuficiente, sin embargo, en muy poco tiempo cientos de personas empezaron a utilizarlo y a pedir más del mismo. A todo lo anterior se sumó que el gran maestro Andi Gutmans comenzó su desarrollo continuo, haciendo que miles de desarrolladores aportaran ideas e incrementando la llamada “Comunidad de Desarrollo de PHP”, que con el objetivo de llevar adelante un lenguaje libre, fácil, rápido y del interés de todos, que fuera muy utilizado por una gran número de internautas, influyendo en que se empezara a tomar en serio.

De esta manera se incrementaron las potencialidades de dicho lenguaje al contar con desarrolladores que, en trabajos continuos, aportaran cualidades y lo fuesen haciendo más grande, llegando a ocupar un lugar entre los lenguajes con más experiencias como Perl, JSP, C#, ASP y ASP.NET. Después de tanto esfuerzo y trabajo se hace una versión llamada PHP3, a continuación para contrarrestar las insuficiencias y los pocos privilegios otorgados para su fácil manipulación surge una versión superior, PHP4. En esta última incorporan algo de la ya conocida programación orientada a objetos (POO), aunque de manera muy abstracta; algo de trabajo con XML, Webservices y seguridad.

Pero les pregunto ¿era esto suficiente? Tiempo después, la demanda de más características orientadas a objetos fue incrementando brutalmente y Andi tuvo la idea de reescribir la parte de lo orientado a objetos en la ingeniería de Zend (Compañía que se dedica al desarrollo estable de PHP y herramientas informáticas para el mismo). A lo que se le fueron adicionando, eliminando y cambiando características según fue confeccionando PHP5. Por ejemplo, se cambió el uso de la herencia múltiple (propio de C++) por las interfaces. Se reescribieron las funcionalidades con XML y se le dio soporte a la librería SOAP y MySQLi.

¿Qué tenía esto de nuevo?

A continuación expongo las características esenciales que trajo consigo esta nueva versión: En el caso de la POO se insertan los modificadores de acceso para métodos y atributos, común en los lenguajes de programación más avanzados, estos son: private, public y protected.

Por ejemplo:

PHP4

```
class Datos{  
    var $nombre = 'Linur'; //así se declara una variable  
    function GetNombre() {  
        return $this>nombre  
    }  
}
```

PHP5

Surgen las llamadas funciones mágicas (a partir de PHP5.2), que como en otros lenguajes tales como C# facilitan el trabajo al desarrollador solo que este mucho más, producto de que en C# se hace para cada atributo, sin embargo aquí en PHP5 es para todos los atributos o propiedades, que cómodo ¿verdad?



```
class Datos{
    private $nombre = 'Linur'; //así es como se declara un atributo
    public function __get($var) {
        //función mágica que sustituye al get clásico
        return $this>$var;
    }
}
```

Los constructores fue otra polémica elaborando un sistema de constructores no semejante a los clásicos conocidos, aunque se mantiene la compatibilidad con el método tradicional para los viejos desarrolladores. Por ejemplo en:

PHP4

```
class Datos{
    var $nombre = 'Linur';
    function Datos( $nombrea ) {
        //declaración de constructor clásico
        return $this>nombre = $nombrea;
    }
}
```

PHP5

```
class Datos{
    private $nombre = Linur;
    public function __construct($var) {
        //función mágica para declarar constructores
        return $this>nombre = $var;
    }
}
```

Se eliminó la herencia múltiple apostándole a las interfaces, las cuales brindan un mayor servicio y comodidad. Las mismas consisten en agrupar en una función un conjunto de funciones que son implementadas en otras clases, las que se unen con un mismo objetivo, dándole al desarrollador una programación más organizada y de mejor entendimiento y legibilidad a la hora de desarrollar aplicaciones, lo cual admite que una clase implemente todas las interfaces que quiera. Por ejemplo:

PHP4

Imposible

PHP5

```
interfaces MostrarNombres(){
    // se utiliza la palabra reservada interfaces
    function Nombres();
}
class Personas implements MostrarNombres{
    // se utiliza la palabra reservada implements
    private $nombre;
    public function Nombres($var) {
        while($var == 'nombre')
            echo $this> nombre;
    }
}
```




Además, permite:

- Definir métodos y clases finales.
- Clonar objetos explícitamente.
- Declarar clases constantes.
- Declarar métodos y atributos estáticos.
- Declarar clases y métodos abstractos.
- Definir una función `__autoload` para cargar otros archivos o páginas.
- Perfeccionar el manejo de errores.

Por otra parte,

- Las antiguas funcionalidades que presentaba PHP4 para el soporte de XML fue usando una variedad de librerías XML, pero ya con PHP5 se volvió implementar usando librerías más potentes y más estándares, como libxml2 (potente librería del proyecto GNOME). Probablemente este haya sido uno de los mayores logros hasta la actualidad de PHP. Además, se creó el SimpleXML que hace del trabajo de XML algo demasiado sencillo.
- La librería SOAP en PHP4 se logró con una implementación en PEARs, pero ya fue incluida como una implementación nativa de PHP, desarrollada en C.
- Se reescribió la extensión para MySQLi permitiendo una mayor configuración en la interfaz OO, dándole una mayor ventaja sobre el antiguo MySQL 4.1 y anteriores.
- Se aumentó las ventajas sobre la extensión SQLite.
- Se incluyó soporte para la librería Tidy, poderosa para parsear entre otras muchas cosas en documentos HTML.
- No se incluye en el paquete de PHP5 la extensión para Perl, pero se puede adjuntar permitiendo la llamada a scripts, usar objetos de Perl y otras funcionalidades nativas desde dentro de PHP.
- Se realizó un nuevo administrador de memorias que libera los bloques de memorias con mucha más rapidez y eficiencia. Factor esencial.
- Se eliminó el soporte para Windows 95 (ojalá fuera para todos), producto de que Microsoft dejó de dar soporte en el 2002 y se decide eliminar el soporte al mismo, aunque cabe decir que PHP está orientado al Software Libre.

Después de una breve explicación de las nuevas características en PHP5, deberíamos analizar entonces que, con las múltiples ventajas que trae consigo, no debemos despreciarlo, sabiendo que mientras más sencillo de entender y potente sea un lenguaje mejor será su aceptación por los programadores.

Esto lleva a lo siguiente y fundamental en este artículo; muchos plantean que aún no migran sus aplicaciones hacia PHP5 porque:

“no es estable”; a lo que respondo: “es estable desde su versión 5.1.6, actualmente en la versión 5.2.2 (momento en que creó este artículo)”

“no es seguro, ya que le fueron encontrados varios huecos ”; a lo que respondo: “a PHP4 le fueron encontrado grandes huecos y brechas de seguridad a lo que hubo que hacer una versión llamada 4.4.7 e invocar que se migraran los servidores porque podían ocurrir ataques violentos, por lo que todo lenguaje y versiones de los mismos presenta huecos y errores que se van solucionando y más cuando existe una mega



comunidad y un proyecto llamado PHP Security dedicado a encontrar vulnerabilidades y parchearlas en un máximo de una semana.”

“es muy joven aún”; y digo de nuevo: “ya lleva 2 años de evolución, sin embargo aceptamos las nuevas versiones de los frameworks de C#, las nuevas Máquinas Virtuales de Java y todos muchos más joven que esta versión de PHP, ¿por qué?, porque PHP es libre, que va compañeros, lo desarrollan y aportan los propios desarrolladores de la Microsoft, Sun Microsystem, Oracle, entre muchos otros, sin contar a la compañía que le da soporte que es la Zend”

Y un punto final para esto sería: **¿qué vamos hacer ahora que se le dejará de dar soporte a PHP4?** ¿Será que estamos esperando la versión 6? Esto sería para reír, porque ¿serán capaces de adaptarse a la nueva versión? En otro artículo veremos como viene esta versión y todo lo que promete, por ahora solo decirles que la mejor alternativa y opción es “andar de la mano con el mundo, en su constante progreso” o “vivir de engaños toda la vida y morir creyendo que estamos bien”.



Figura 19. PHP5, uno de los lenguajes de programación más populares y extendidos de Internet²³

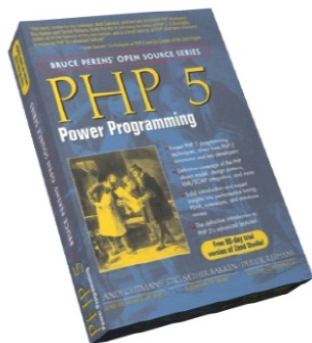


Figura 20. Libro PHP5 Power Programming (720 páginas), Andi Gutmans²⁴

²³ www.pc-cuadernos.com/newsite/booklet.php?id=68

²⁴ http://www.phptr.com/content/images/013147149X/downloads/013147149X_book.pdf



How To GDM

Yunier Vega Rodríguez
yvrodriguez@estudiantes.uci.cu
Proyecto UNICORNIOS

¿A cuántos, usuarios comunes de Microsoft Windows, no les habrá llamado la atención al enfrentarse por primera vez a la manera tan distinta en que los Sistemas GNU/Linux bootean, inicializan su sesión o permiten que usted simplemente acceda a ella? La variedad es tanta que muchos, aventurados en sus secretos, han querido descifrar el misterio de la interfaz con que se nos presenta, algo que ha sido considerado erróneamente hasta hace poco sólo una consola. En este artículo se presenta el cómo hacer un GDM, extensible también a toda la familia de gestores de sesión que conocemos como XDM y KDM, salvo por algunas diferencias.

GDM

Gnome Display Manager, por sus siglas en inglés, es el Administrador del entorno de escritorio Gnome, que gestiona la configuración de inicio de sesión para los usuarios en los Sistemas GNU/Linux. Un tema de GDM, definido como la interfaz del GDM, es un compactado (.tar.gz) de una carpeta que contiene:

- Reconocimiento de la Interfaz por el Sistema (GdmGreeterTheme.desktop)
- Diseño de la interfaz(.xml)
- Imagen background (.png, .jpg, ...)
- Imagen Screenshot (.png, .jpg, ...)
- Imágenes de botones (.png, .jpg, ...)
- Imagen prompt (.png, .jpg, ...)

Editar GdmGreeterTheme.desktop

Para llevar a cabo esta tarea se necesita primeramente abrir la consola, luego, empleando un editor cualquiera modificar los siguientes parámetros:

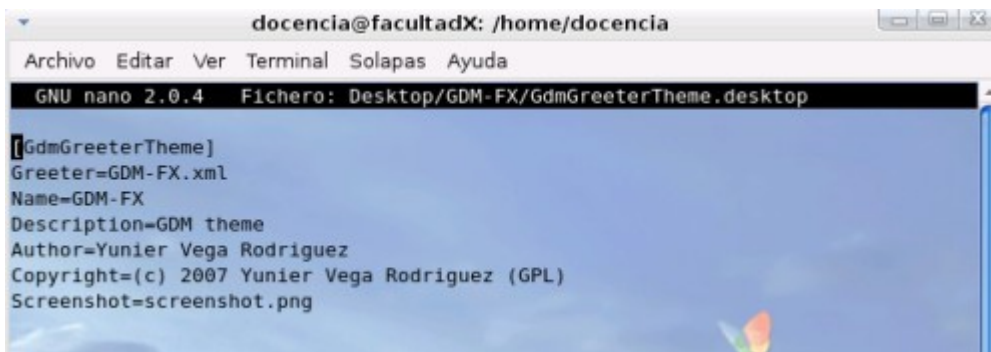


Figura 21: Ejemplo del fichero GdmGreeterTheme.desktop a través de una consola

Greeter: aquí colocar el nombre del fichero xml. Ejemplo: TemaGDM.xml.

Name: Nombre del tema

Description: descripción del tema; por lo general esta no varía, de manera que no interesa si le dejamos la que trae por defecto: GDM theme.

Author: aquí escribir el nombre del autor del tema.

Copyright: derecho de autor del tema: Sólo escribir el nombre del autor y el año, el resto puede dejarse tal y como está. Ejemplo: (c) 2007 Juana de Arcos (GPL).



Screenshot: nombre de la imagen del tema: es una imagen que muestra cómo se vería el tema una vez instalado, la cual por principio puede tener cualquier formato de imagen conocido, es decir png, jpg, etc., pues la extensión es modificable en el Greeter.

Editar el fichero .xml

Debido a que las funcionalidades de un GDM, - al menos las que conciernen al objeto de la interfaz y no a su diseño-, son funcionalidades, que de forma general ya están implementadas, nuestro trabajo consistirá en diseñar una interfaz de GDM basados en XML.

XML esencialmente define la manera particular en que serán tratados los componentes del GDM y no su diseño gráfico, para el que podemos auxiliarnos de herramientas como GIMP o Blender. A continuación se expone la manera en que puede realizarse esto para las opciones más importantes que provee el GDM.

Componentes:

- **Background**

En este item se define el nombre de la imagen background del tema, así como la posición a ocupar por esta, es decir:

type="pixmap": indica que es una imagen.

x="" y="": posición del item.

file="background.png": indica que la imagen a emplear para el item(background del tema) es background.png.

width="" height="": definen el tamaño del item.

```
<!-- Imagen Background -->
<item type="pixmap">
  <normal file="background.png"/>
  <pos y="0" x="0" width="100%" height="100%"/>
</item>
```

- **Loginbox**

Para este caso las propiedades son similares al anterior, la única diferencia está en la imagen a emplear: loginbox.png.

```
<!-- Imagen Prompt background -->
<item type="pixmap">
  <normal file="loginbox.png"/>
  <pos x="50%" y="50%" width="1024" height="768" anchor="c"/>
</item>
```

- **Prompt de logueo**

En este item se definen internamente dos item anidados , cada uno con una funcionalidad distinta. Veamos:

type="rect": el tipo rect define un componente similar a un div en html o layer o capa en Dreamweaver.

type="label": como se puede intuir de su nombre, el tipo "label" se emplea para escribir textos que serán mostrados en pantalla. La funcionalidad de este item para el GDM consiste en indicarle al usuario en que entry deberá autenticarse para acceder a su sesión.

type="entry": este tipo define que el item será un edit, o caja de texto.

id="": podemos definirlo de manera similar a como lo hacíamos en html. La función del id no es otra que identificar el item para su posterior manipulación.

x="" y="": si nos referimos a los item internos a "rect", entonces las posiciones y y x se establecen de acuerdo a las dimensiones de este, más si lo hacemos para "rect", la posición será respecto a las dimensiones de la pantalla ,pues "rect" no está contenido en ningún otro item.



```
<!-- Logueo de usuario -->

<item type="rect">
  <pos x="0" y="300" width="1024"/>

  <box orientation="horizontal">

    <!-- username label -->
    <item type="label" id="pam-prompt">
      <pos x="180" y="90" anchor="nw"/>
      <normal font="Sans Bold 10" color="#000000"/>
      <stock type="username-label"/>
    </item>

    <!-- user-pw-entry -->
    <item type="entry" id="user-pw-entry">
      <pos x="212" y="90" width="172" height="19"/>
      <normal font="Sans 10" color="#000000"/>
    </item>
  </box>
</item>
```

- **Botones**

Sesión:

Existen muchas maneras, como se verá de diseñar un botón. La más común de las formas no tiene nada que ver con el xml, pues lo que hace a un botón lo que es, es su funcionalidad, la cual puede definirse empleando para ello un lenguaje de programación de escritorio como C++/GTK o Python/GTK, por ejemplo. Una imagen o un texto, pueden constituir, por tanto un botón, de manera que en xml, cualquiera de los tipos que muestren al usuario algo, cumplen con este concepto. En XML existen tres tags que de forma general ayudan a definir algunos de los eventos de un botón, estos son, de acuerdo con el código mostrado arriba:

<normal>: que define el estado inicial o por defecto en el que se mostrará el item.

<prelink>: establece el estado en el que se mostrará el item una vez se haya pasado el mouse por encima de este.

<active>: establece el estado del item después de cliqueado.

En el código anterior, lo que define al botón, no es precisamente el tipo del item, el cual es un layer, sino el id con el cual habrá de establecerse esta funcionalidad.



```
<!-- Boton Sesion -->
<item type="rect" id="session_button" button="true">
  <pos x="320" y="430" width="100%" height="box" anchor="nw"/>
  <box spacing="5" xpadding="3" ypadding="2" orientation="horizontal">
    <item type="pixmap">
      <pos y="50%" width="15" height="15" anchor="w"/>
      <normal file="session.png"/>
      <prelight file="session-prelight.png"/>
      <active file="session-active.png"/>
    </item>
    <item type="label">
      <pos y="50%" width="50" height="20" anchor="w"/>
      <normal color="#f0f0f0" font="Bitstream Vera Sans Bold 10"/>
      <prelight color="#ecddb5" font="Bitstream Vera Sans Bold 10"/>
      <active color="#ecddb5" font="Bitstream Vera Sans Bold 10"/>
      <stock type="session"/>
    </item>
  </box>
</item>
```

Idioma:

El botón de idioma se define de manera similar al anterior, la única diferencia está en el id empleado para el item de este.

```
<!-- Boton Lenguaje -->
<item type="rect" id="language_button" button="true">
  <pos x="360" y="458" width="100%" height="box" anchor="nw"/>
  <box spacing="5" xpadding="3" ypadding="2" orientation="horizontal">
    <item type="pixmap">
      <pos y="50%" width="15" height="15" anchor="w"/>
      <normal file="language.png"/>
      <prelight file="language-prelight.png"/>
      <active file="language-active.png"/>
    </item>
    <item type="label">
      <pos y="50%" width="50" height="20" anchor="w"/>
      <normal color="#f0f0f0" font="Bitstream Vera Sans Bold 10"/>
      <prelight color="#ecddb5" font="Bitstream Vera Sans Bold 10"/>
      <active color="#ecddb5" font="Bitstream Vera Sans Bold 10"/>
      <stock type="language"/>
    </item>
  </box>
</item>
```

Acciones:



Igual que el componente anterior.

```
<!-- Boton Acciones -->
<item type="rect" id="system_button" button="true">
  <pos x="415" y="480" width="100%" height="box" anchor="nw"/>
  <show modes="console" type="system"/>
  <box spacing="5" xpadding="3" ypadding="2" orientation="horizontal">
    <item type="pixmap">
      <pos y="50%" width="15" height="15" anchor="w"/>
      <normal file="system.png"/>
      <prelight file="system-prelight.png"/>
      <active file="system-active.png"/>
    </item>
    <item type="label">
      <pos y="50%" width="50" height="20" anchor="w"/>
      <normal color="#f0f0f0" font="Bitstream Vera Sans Bold 10"/>
      <prelight color="#b4cee6" font="Bitstream Vera Sans Bold 10"/>
      <active color="#b4cee6" font="Bitstream Vera Sans Bold 10"/>
      <stock type="system"/>
    </item>
  </box>
</item>
```

GDM y Diseño

Como bien se establece en los apartados precedentes, el proceso de construcción de un GDM es, a todas luces un proceso de creación artística más que de programación, donde los usuarios pueden incluso echar a volar su imaginación a fin de adecuar su espacio de trabajo a sus preferencias y gustos estéticos.



GNU/Linux en los CDI: Una experiencia libre

Yordan Vélez Figueredo
luckymanfar@gmail.com
Universidad de las Ciencias Informáticas

Muchos médicos cubanos cumplen misión en los CDI (Clínicas de Diagnóstico Integral) venezolanos. Durante meses especialistas sirven como técnicos de laboratorio, informáticos o administradores de redes. Muchos cubanos de diferentes edades han participado en esta linda misión - donde se ha hecho realidad el milagro - durante los últimos dos años. A pesar del enorme esfuerzo que representa este sistema integral gratuito de salud, con altas tecnologías de hardware y software, es poco probable encontrar un sistema operativo GNU/Linux, pues aún nuestros profesionales realizan la mayor parte de su formación sobre plataformas privativas.



Figura 22: Foto tomada al CDI "Salvador Allende"

Punto de partida de la investigación...

La mayor parte de los que utilizan los medios informáticos en un CDI son médicos y enfermeras. Es muy común que estos necesiten utilidades y herramientas propias de los usuarios comunes, algo que a los especialistas nos resulta bastante difícil de comprender acostumbrados a las consolas y los lenguajes de programación abstractos.



Figura 23: Doctora utilizando una PC con GNU/Linux



Un CDI con GNU/Linux

Tengo instalado GNU/Linux en mi PC de trabajo del CDI; todo el mundo me calificó de loco porque no podría hacer las cosas que normalmente debería hacer como parte de la misión. Después de unos días de muchas configuraciones y lecturas en internet, logré tener mi PC lista y funcional. Como era de esperar, le instalé Beryl, para que las personas vieran algo gráfico de GNU/Linux, y se enamoraran de él.



Figura 24: El autor con una distro GNU/Linux en su PC

Logré así que varios médicos abandonaran el imperio privativo aunque no del todo, porque la distribución que les he instalado es Xubuntu Feisty Down. Los médicos solicitan continuamente cosas para que uno les ayude a hacer, y entre dichas tareas se encuentra convertir videos en formato AVI, para DVD, se me ocurrió buscar si se podría hacer esto con programas libres y encontré una solución complicada en internet, en un foro de Ubuntu, y que fue reproducida en el portal de la UCI²⁵.

Una solución tan compleja puede ocasionar muchos retrasos. Decidí utilizar algunos recursos de consola que también estaban presentes en la primera solución, e implementarlo en pocos pasos.

¿Cómo crear DVD/VCD de video a partir de un .avi y un .srt?

Hace poco sugerí una serie de pasos para crear un disco DVD/VCD a partir de un fichero de extensión avi y su respectivo fichero de subtítulos. Debido a lo extensa y algo confusa que puede ser dicha tarea, hago una nueva propuesta de comandos, mucho más sencilla y óptima, además ya he creado un script que automatiza la labor.

Para nuestro trabajo necesitamos tener instalado el **mencoder**, que lo pueden instalar según las especificidades de cada distribución de GNU/Linux. Una vez instalado el programa, en un fichero de texto, escribimos lo siguiente:

```
#!/bin/bash
mencoder ovc lavc oac mp3lame $1 o $2.avi sub $3
mencoder $2.avi of mpeg ovc lavc lavcopts vcodec=mpeg1video:
vbitrate=1100 ofps 25 oac lavc lavcopts acodec=mp2:arbitrate=128 srate
44100 vf pp=hb/vb/dr/al/1k o $2.mpg
```

Guardamos el archivo con el nombre que deseemos, por ejemplo *mpg2dvd.sh* en **/usr/bin** y le damos los permisos suficientes para que cualquiera pueda ejecutarlo. Ahora sólo nos resta mandar a ejecutar el script y tener un tin de paciencia mientras se convierte el video.

²⁵ <http://softwarelibre.uci.cu/modules/news/article.php?storyid=2515>



mpg2dvd.sh fichero_original.avi fichero_salida fichero_subtitulo.srt

fichero_original.avi: es el nombre (incluyendo la extension) del archivo original.

fichero_salida: es el nombre (sin extension) que desea darle al fichero de salida.

fichero_subtitulo.srt: es el nombre del fichero que tiene el subtitulaje

Cuando esté listo el archivo fichero_salida.mpg podemos quemarlo con Gnomebaker, K3B, etc... que cualquiera de ellos ofrece interfaz gráfica fácil de usar.

Conclusiones

Si se compara la primera solución encontrada a la segunda es muy fácil comprender que podemos lograr importantes avances cuando investigamos. Aún más si se compara con la opción "0" de usar Windows, y rehusar las variantes de GNU/Linux. Algunos recomiendan también usar aplicaciones como DeVeDe, pero a veces suele no llegar una información tan valiosa cuando nos encontramos aislados, e internet nos inunda con su redundancia. Siempre recomiendo investigar las mismas.

El uso de software libre hoy en el mundo se ve frenado sólo para los usuarios finales, que muchas veces deben pasar mucho trabajo para lograr cosas que en Windows son simples. En gran medida sucede por la no orientación de los sistemas GNU/Linux a ámbitos y fines comerciales. Es cambiable; sin embargo la voluntad con que enfrentamos la tarea de hacer a nuestros países independientes tecnológicamente.



Figura 26: Fachada del CDI "Salvador Allende"



Figura 25: Instalación de Debian en la PC del autor



“Los momentos más importantes del SWL en nuestro país son los que estamos viviendo”

*Entrevista a Yoel Benítez Fonseca
Por Ing. Abel Meneses Abad
3 de Octubre 2007*

Yoel Benítez Fonseca, oriundo de Niquero, es actualmente el administrador de la red de la Facultad Territorial de la UCI en Granma. Graduado de la Universidad de Oriente en el año 2002, de Licenciatura en Ciencias de la Computación. Ha participado a lo largo de 9 años en las comunidades cubanas de software libre.

¿Cuándo fue la primera vez que supo de la existencia del Software Libre en Cuba y desde cuando lo utiliza?

“Conscientemente, a partir del 98, digo consciente porque antes de eso más o menos a partir de 1er año de la carrera (cibernética) a mediados del año 96, ya estaba usando SWL.”

¿Utiliza Software Libre en su centro de trabajo? ¿Desde cuándo?

Bueno, aquí en la FRG todo es relativamente nuevo y la facultad completa ha migrado a SWL, así que uso el SWL para todo. Por otro lado, como soy administrador de red desde que inicie la vida laboral siempre he estado ligado de una forma u otra a el SWL, esos serian unos 5 años de experiencia después de graduado.

¿Qué sistema operativo libre utilizó por primera vez?

“RedHat Linux 5.0 y el Debian de aquellos tiempos que no recuerdo bien que versión era, del RedHat si estoy seguro porque para ese tiempo yo era miembro de la comunidad LINUX-STGO y recibimos una copia original de esa distribución cada uno de los miembros.”

¿Conoce las primeras distribuciones de GNU/Linux que se utilizaron en nuestro país? Cuales? ¿Cuándo se utilizaron por primera vez?

“Bueno, como mencioné antes creo que RedHat y Debian fueron las primeras, también algo de SuSE pero no estoy seguro de ninguna de las 3 porque aunque soy viejo en esto no me considero un pionero del SWL en Cuba. Si sé que cuando me integré a la comunidad de Santiago, RedHat y Debian eran muy populares. Y no sé en realidad cuándo fue la primera vez que se usó SWL en Cuba, pero creo que los primeros fueron los de INFOMED.”

¿Cuáles son las instituciones que usted conoce utilizan tecnología o software libre? ¿Desde cuándo lo utilizan?

“Que yo sepa, las universidades, y cuando digo las universidades estoy incluyendo todos los centros de educación superior del país. Y de nuevo no puedo contestar cuando porque eso sencillamente no lo sé.”

¿Qué personas conoce son importantes en la historia del SWL en Cuba?

“Bueno, yo conocí en un tiempo a Omar I. Marquez Merceron, él fue el culpable de que yo haya conocido algo del SWL. No sé si será una persona importante en la historia del SWL en Cuba, pero en el tiempo en que lo conocí (estoy hablando de 1996..1998) se podía decir que era un Guru en la materia. Otro que conocí fue a Adriano Galano Díaz que actualmente reside en España y que era uno de los principales cabecillas del grupo LINUX-STGO. El grupo era bastante grande pero creo que esos eran los de más influencia en la comunidad santiaguera. En cuanto al resto del país no sabría decir nombres.”



¿Que distribuciones GNU/Linux ha usado?

“¡Esa lista!, aquí van: RedHat, SuSE, Debian, Knoppix, Fedora, Gentoo, Ubuntu, Kubuntu, LFS (no sé si cuenta como distribución), es posible que se me quede alguna.”

¿Ha recibido o impartido algún curso?

“Recibido no, impartido si, como parte de la docencia en la Universidad, casi todos cursos básicos de sistemas operativos, excepto uno para los estudiantes de Informática en la Universidad de Granma sobre cómo hacer manejadores de dispositivos en Linux.”

¿Cuáles fueron los momentos más importantes a su juicio en el desarrollo del Software Libre en nuestro país?

“Creo que los momentos más importantes del SWL en nuestro país son los que estamos viviendo, porque creo que ahora es que se esta tomando un poco más en serio, como una política el uso del software libre y la posibilidad de que Cuba se convierta en una potencia en este sentido. Antes, se han realizado esfuerzos por parte de individuos e instituciones que han visto el potencial del SWL, pero realmente creo que ahora es el momento justo para que el país migre completamente a SWL...”

¿Si tuviera que construir la historia del Software Libre en el ámbito en que usted se desarrolla cómo la escribiría?

Bueno, yo comenzaría por buscar en Internet los archivos de las listas de discusión y comenzaría a escribir a las personas que aparecen (si mantienen esas mismas direcciones todavía). Otra idea puede ser leer los archivos de las listas, leyéndolos es posible armar la historia de ese grupo o comunidad. Actualmente existe una lista de discusión llamada linux-l²⁶, que tiene un volumen diario de correos bastante pesado, a veces de 100 mensajes diarios, en la que participan algunas de las personas con más experiencia en el SWL del país, esa podría ser otra de las vías.

Entrevista realizada por el Ing. Abel Meneses Abad, durante las labores de migración de la Facultad Territorial de Granma de la UCI a software libre, utilizando la distribución Fedora 7. Esta entrevista está enmarcada en el estudio que realiza un grupo de estudiantes de la Universidad de las Ciencias Informáticas (UCI), que investigan la Historia del Software Libre en Cuba, el resultado final consistente en una página Web sobre el tema será presentado en el Concurso Internacional sobre Historia de la Informática patrocinado por la IEEE con sede en Washington, EE.UU. Ellos han dividido la investigación en 3 etapas fundamentales: 1984-1992, 1992-2002, 2002-2006. Parte de la recopilación de información se produce mediante este tipo de entrevista.

²⁶ <http://listas.softwarelibre.cu/mailman/listinfo/linux-l>



Construyendo una terminal

Eiger Mora Moredó
emora@estudiantes.uci.cu
Proyecto UNICORNIOS

Por muchos es conocida la importancia que para los sistemas GNU/Linux tienen los terminales o consolas, que en no pocos casos le han merecido injustamente el calificativo de “sistemas de palo”. Lo cierto es que, sin quitarle o agregarle méritos, la consola es un arma fundamental para todo el que quiera conocer a fondo el terreno que pisa, para todo el que quiera en verdad conquistar su sistema.

Sobre este tema existe una duda recurrente entre quienes hoy visitan los foros de nuestro portal de Software Libre. En esta edición les presento cómo construir un terminal utilizando C, Gtk y la librería VTE (Virtual Terminal Emulator) con la cual se puede construir un widget “terminal” para Gtk; el IDE usado es Anjuta; Sistema Operativo, Debian GNU/Linux 4.0.

Instalar librerías

Antes de comenzar, revisemos si tenemos instaladas las librerías necesarias:

```
libgtkmm-2.4-dev  
libglademmm-2.4-dev  
libvte-dev  
libgtkmm-2.4-doc  
libglademmm-2.4-doc  
libvte-doc
```

Nota: Se incluyen los paquetes de documentación para tener la referencia de la librería, se recomienda instalar el Devhelp.

Creación del Proyecto

Primeramente creamos un proyecto “Genérico de consola” para C en nuestro IDE, seguidamente configuramos las opciones de compilación, para esto nos dirigimos a la Barra de Menú y escogemos: **Opciones -> Opciones del Compilador y Enlazador**. En la pestaña Opciones definimos las CFLAGS y LDFLAGS como muestra la siguiente imagen.

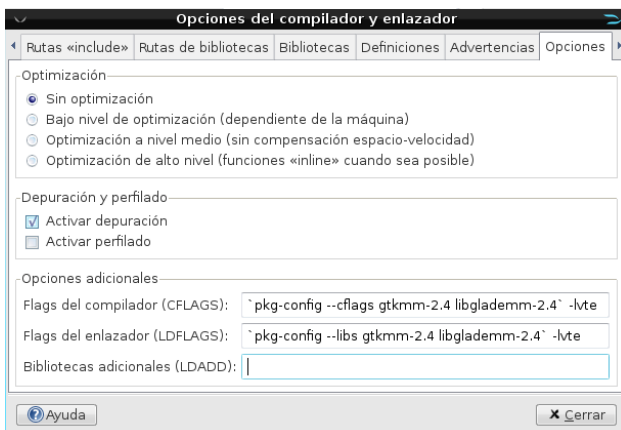


Figura 27: Pestaña de Opciones en Opciones del Compilador y el Enlazador



Generando código

Luego de los primeros pasos, todo está listo para comenzar con el código, siendo así vamos sobre el archivo main.c

```
//incluyendo librerías
#include <gtk/gtk.h>
#include <vte/vte.h>

int main(int argc, char* argv[])
{
    // GtkWidget es el tipo de dato usado para los widgets
    GtkWidget* window;
    GtkWidget* terminal;
    /* En cualquier aplicación hay que realizar la siguiente llamada. Los argumentos son
    tomados de la línea
    * de comandos devueltos a la aplicación. */
    gtk_init(&argc,&argv);
    // construyendo la ventana principal
    window=gtk_window_new(
        GTK_WINDOW_TOPLEVEL
    );
    // Poniéndole título a la ventana principal
    gtk_window_set_title (GTK_WINDOW
        (window),
        "Terminal");
    // Dándole la dimensión inicial a la ventana principal(800x600)
    gtk_widget_set_size_request(GTK_WIDGET
        (window),
        800,600);
    //Construyendo el componente Terminal
    terminal=(GtkWidget*)
        vte_terminal_new();
    // Definiendo la terminal que el widget va a emular(xterm)
    vte_terminal_set_emulation(
        VTE_TERMINAL
        (terminal),
        "xterm");
    // Escondiendo el mouse
    vte_terminal_set_mouse_autohide(
        VTE_TERMINAL(terminal),
        TRUE);
    // Limpiando la terminal
    vte_terminal_reset(VTE_TERMINAL
        (terminal), TRUE,
        TRUE);
    // Comando a ejecutar en la terminal
    vte_terminal_fork_command(VTE_TERMINAL
        (terminal),
        "/usr/bin/bash",
        NULL,NULL,"~/",
        FALSE, FALSE,
        FALSE);
    // Añadiendo widget terminal al widget contenedor window
    gtk_container_add(GTK_CONTAINER
```




```
        (window), terminal);  
// Mostrando los widgets  
gtk_widget_show_all(window);  
// Bucle de Gtk en espera de eventos  
gtk_main();  
return 0;  
}
```

Se compila el programa y listo, ya tendremos nuestra terminal. Ahora bien, la terminal que se obtiene es bastante primitiva – ni los colores podemos cambiar – claro, quedaría implementarle las potencialidades que desee cada cual.



Sistema de Almacenamiento Distribuido

Daniel Marino Mirayes Taset
dmirayes@estudiantes.uci.cu
Proyecto JDCS & GRID, Polo de Bioinformática

Resumen

La solución para hacer almacenamiento distribuido que se presenta a continuación es una de las investigaciones más populares en estos momentos en el entorno universitario UCI; junto a la ya ejecutada solución del /home compartido, para tener usuarios en un entorno controlado de proyecto, y donde estos puedan sentarse en cada computadora disponiendo siempre de sus documentos sin notar el cambio físico de lugar.

Cada día se abren posibilidades de aprovechar la capacidad de almacenamiento perdida en dicha solución. Algunos proyectos ejecutan acciones ambiciosas de trabajo en equipo y colaborativo, utilizando servidores centralizados de componentes, documentación e incluso música y video.

El horizonte de la producción se abre hoy con estas soluciones para dar a cada proyecto, o polo productivo un entorno más sencillo de gestionar, y una cultura de servicios que marca hoy la próxima meta a alcanzar si nos comparamos con países avanzados. UXi, la revista digital, publicará estas soluciones en los números siguientes, en esta ocasión y para remarcar su importancia aparece como artículo principal.

Introducción

Debido a lo importante que es para los polos productivos utilizar el espacio ocioso que tienen las PC en los laboratorios de producción, a nuestro grupo se le dio la tarea de investigar la manera de utilizarlo de manera fiable y eficiente. Proponemos una configuración a continuación que combina las tecnologías de NFS (Network File System) y RAID (Redundant Array of Independent Disks), para obtener un gran volumen de almacenamiento y que en un futuro se pueda integrar al SE de la Grid o a cualquier necesidad del Polo.

La manera clara de utilizar la capacidad ociosa de las PC de nuestros laboratorios es utilizar un sistema que permita compartir el espacio de manera segura y eficiente y otro sistema para garantizar la fiabilidad de la información, de manera que si un número razonable de máquinas no están conectadas se pueda seguir trabajando normalmente, pero que además garantice que no se pierda la información en caso de fallo en los discos. Para hacer esto se propone la combinación de las tecnologías RAID y NFS, o como se le nombra: *RAID over NFS*.

Analicemos entonces en qué consiste cada tecnología

La tecnología RAID fue definida por primera vez en 1987 por un grupo de informáticos de la Universidad de California, Berkeley. Este grupo estudió la posibilidad de usar dos o más discos que aparecieran como un único dispositivo para el sistema. En 1988, los niveles RAID 1 a 5 fueron definidos formalmente por David A. Patterson, Garth A. Gibson y Randy H. Katz en el ensayo "*Un Caso para Conjuntos de Discos Redundantes Económicos (RAID)*" (*A Case for Redundant Arrays of Inexpensive Disks (RAID)*), publicado en la Conferencia SIGMOD de 1988 (págs. 109-116) El término «RAID» se usó por vez primera en este ensayo, que dio origen a toda la industria de los conjuntos de discos.²⁷

¿Para qué sirve?

Así pues una RAID sirve para crear un único volumen lógico, el cual físicamente esté compuesto por varios discos físicos o lógicos. Dependiendo de que modo de RAID se utilice, esto servirá para conseguir simplemente

²⁷ Tener en cuenta que al hablar de Software RAID, por discos debe entenderse particiones.



un volumen de capacidad mayor, o para conseguir un volumen con mayor seguridad contra fallos de hardware de los discos que lo componen gracias al almacenamiento redundante de estos. De toda la familia de RAID existente se seleccionó la RAID 5, por las posibilidades que el mismo brinda.

RAID5

- Este es quizás el modo RAID más útil cuando uno desea combinar un mayor número de discos físicos y todavía conservar redundancia. RAID5 se puede usar sobre 3 o más discos. El tamaño del dispositivo RAID5 resultante será $(N-1)*S$, tal y como sucede con RAID4. La gran diferencia entre RAID5 y RAID4 es que la información de paridad se distribuye uniformemente entre los discos participantes, evitando el problema del cuello de botella del RAID4.
- Si uno de los discos falla, todos los datos permanecerán intactos, gracias a la información de paridad. Si dos discos fallan simultáneamente, todos los datos se perderán. RAID5 puede sobrevivir a un fallo de disco, pero no a dos o más.
- El rendimiento de lectura de RAID5 es equiparable al de RAID0 con el mismo número de discos. Exceptuando los bloques de paridad, los cuales pueden causar una ligera ralentización en las escrituras (en las lecturas no se usan los bloques de paridad a no ser que algún disco falle).

NFS

- El sistema NFS está dividido al menos en dos partes principales: un servidor y uno o más clientes. Los clientes acceden de forma remota a los datos que se encuentran almacenados en el servidor. Las estaciones de trabajo locales utilizan menos espacio de disco debido a que los datos se encuentran centralizados en un único lugar pero se puede acceder a la información y modificarla por varios usuarios, de tal forma que no sea necesario replicar la información.
- Los usuarios no necesitan disponer de un directorio “home” en cada una de las máquinas de la organización. Los directorios “home” pueden crearse en el servidor de NFS para posteriormente poder acceder a ellos desde cualquier máquina a través de la infraestructura de red.
- También se pueden compartir a través de la red dispositivos de almacenamiento como disqueteras, CD-ROM y unidades ZIP. Esto puede reducir la inversión en dichos dispositivos y mejorar el aprovechamiento del hardware existente en la organización.

Todas las operaciones sobre ficheros son síncronas. Esto significa que la operación sólo retorna cuando el servidor ha completado todo el trabajo asociado para esa operación. En caso de una solicitud de escritura, el servidor escribirá físicamente los datos en el disco, y si es necesario, actualizará la estructura de directorios, antes de devolver una respuesta al cliente. Esto garantiza la integridad de los ficheros. La combinación que proponemos es utilizar RAID sobre NFS, permitiendo que las particiones hechas en las máquinas clientes puedan ser utilizadas por el servidor utilizando dispositivos virtuales de loopback.

Instalación

Cliente

Primeramente se explicará cómo configurar las PC que van a unirse al SAD.

Debe de instalarse el paquete nfs:

```
$ sudo aptitude install nfs
```

Crear un directorio, en este caso llamado sadspace, en la partición raíz:

```
$ sudo mkdir /sadspace
```

Luego se exporta la carpeta creada, de manera tal que se pueda acceder a ella desde el servidor:

```
$ sudo echo "/sadspace 10.7.19.0/24(rw,async,no_root_squash)" >>etc/exports
```

De esta manera se permite que pueda accederse a este recurso desde cualquier dirección del rango dado.
Ahora se reinicia el servicio de NFS

```
$ sudo /etc/init.d/nfs restart
```

Servidor

Ahora se explicarán las configuraciones que se deben hacer en el servidor. Inicialmente se debe instalar el paquete *mdadm* y todas sus dependencias. En Ubuntu viene instalado por defecto.

```
$ sudo aptitude install mdadm
```

Se crean tantos puntos de montajes como clientes se tengan, en este caso 3.

```
$ sudo mkdir /mnt/client1
```

```
$ sudo mkdir /mnt/client2
```

```
$ sudo mkdir /mnt/client3
```

Se añade al fichero de configuración fstab (/etc/fstab) la forma en que se montarán las particiones:

```
client1:/sadspace /mnt/client1 nfs rw,user,defaults,umask=000 0 0
```

•

•

•

```
clientN:/sadspace /mnt/clientN nfs rw,user,defaults,umask=000 0 0
```

Se montan las particiones

```
mount -a
```

Luego se pasa a restringir la capacidad a utilizar. Es evidente que al crear la carpeta en las máquinas clientes, se podría utilizar cualquier cantidad, pero debe de restringirse este espacio. En este ejemplo se restringe para 5GB (5000MB) por clientes. Ejecutar entonces como root:

```
dd if=/dev/zero of=/mnt/client1/container bs=1M seek=4999 count=1
```

•

•

•

```
dd if=/dev/zero of=/mnt/clientN/container bs=1M seek=4999 count=1
```

Ahora se crearán los dispositivos virtuales. Para que pueda ser configurado correctamente después el RAID:

```
$ sudo losetup /dev/loop1 /mnt/client1/container
```

- • \$

```
$ sudo losetup /dev/loopN /mnt/clientN/container
```

Configuración del RAID

El primer paso será preparar las particiones a utilizar para crear la RAID. Para esto se puede utilizar cualquier herramienta de particionado. En el presente caso se utilizará GParted, una herramienta gráfica de fácil uso. Puesto que Ubuntu no la lleva instalada por defecto se instalará mediante el Gestor de paquetes Synaptic, o bien con apt-get mediante comandos:

```
$ sudo apt-get install gparted
```

Una vez instalado GParted, se procederá a crear una partición para la totalidad de cada uno de los discos idénticos de los que se dispone para crear la RAID. Se debe crear una **partición sin formato**, puesto que el formato de la RAID le será dado cuando esta esté construida. A parte del formato se indicará que se trata de un disco para crear una RAID. Esto puede ser hecho de manera fácil con GParted, seleccionando la partición y dirigiéndose al Menú Partición>gestionar señaladores y marcando el señalador "RAID", tal y como muestran las imágenes.

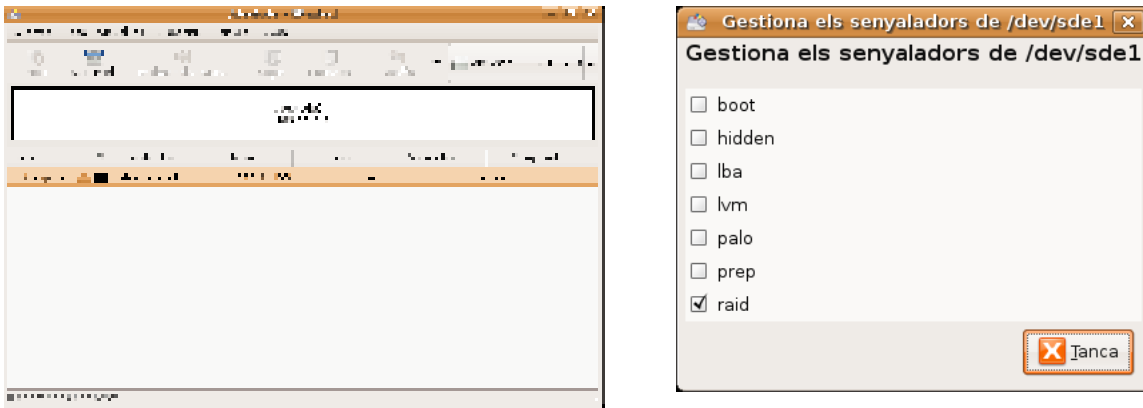


Figura 28: GParted con partición sin formato y señalador de RAID

Marcar la partición como RAID no es vital para el funcionamiento de la RAID, asimismo es la manera más correcta de hacerlo y ayudará a distinguir las particiones en un futuro. Con `fdisk` se puede ver este hecho marcado como "Autodetección Linux raid" (*Linux raid autodetect*). Una vez acabado el proceso podrá ser visto con `fdisk -l` un listado de las particiones como el siguiente (esta en Català debido a que la bibliografía consultada está en ese idioma).

Creación del RAID

Antes de crear el RAID podrá verse en el fichero `/proc/mdstat` si al algún otro array:

```
$ sudo cat /proc/mdstat
Personalities :
Event: 0
unused devices: <none>
```

Una vez establecido que no aparece ningún array. Se proseguirá a la creación del md en el que se creará el RAID. Para ello se utilizará el comando **mknod** como se muestra en la siguiente orden:

```
$ sudo mknod /dev/md0 b 9 0
```

Si ya se tuviera algún otro array llamado md0, puede crearse un md diferente: md1, md2, ... Se procederá a crear finalmente el RAID:

```
$ sudo mdadm create /dev/md0 level=raid5 raiddevices=4 /dev/loop1 /dev/loop2 /dev/loop3 ...
/dev/loopN
Continue creating array? yes
mdadm: array /dev/md0 started
```

Una vez enviada la orden, los discos comenzarán a crear la RAID. Este proceso puede durar varias horas dependiendo de la capacidad de los discos y la potencia del ordenador/discos. Se podrá visualizar en cualquier momento el estado de éste proceso en el fichero `/proc/mdstat`:

```
$ cat /proc/mdstat
Personalities : [raid5] [raid4]
md0 : active raid5 sdb1[0] sde1[3] sdd1[2] sdc1[1]
586075008 blocks [4/3] [UUU_] [ >..... ] resync = 0.7% (4103401/586075008)
finish=177.6min speed=97640K/sec
```

[4/3] [UUU_] indica el número de discos activos y correctos en ese momento. No deberá preocuparse porque durante éste periodo de creación de la RAID nos marque que hay alguno incorrecto. Cuando finalice el proceso éste indicador deberá mostrar que todos los discos están correctos. Una vez concluido el proceso de construcción de la RAID podrá ser examinado el fichero `/proc/mdstat` el estado de nuestro array:



```
$ cat /proc/mdstat
Personalities : [raid5] [raid4]
md0 : active raid5 sdb1[0] sde1[3] sdd1[2] sdc1[1] 586075008 blocks level 5, 64k chunk,
algorithm 2 [4/4] [UUUU]
unused devices: <none>
```

¡Ya tenemos creada nuestro RAID 5 distribuido! Una vez creado solo debe ser montada y formateado con el sistema de ficheros que se desee.

Formatear el RAID

Para formatear se utilizará el comando **mkfs**:

Con ReiserFS

```
mkfs.reiserfs /dev/md0
```

Con ext3

```
mkfs.ext3 /dev/md0
```

Montar el RAID

Para montarlo añadimos la siguiente línea al fichero */etc/fsta*:

```
/dev/md0 /punto_de_montaje sistema_de_archivos defaults,user 0 0
```

Recordar que debe tenerse creada la carpeta **/punto_de_montaje** (la ruta deseada), con los permisos correspondientes a los usuarios que se quiera que accedan a l RAID. También debe especificarse qué sistema de archivos es el RAID cambiando **sistema_de_archivos** por **reiserfs**, **ext3**...

Para montar el RAID bastará luego con hacer:

```
$ sudo mount /punto_de_montaje
```

O bien reiniciar y que Ubuntu lo monte automáticamente en el inicio del sistema.

Monitorización del estado un RAID y sus discos

Para ver el estado actual de los discos y unidades RAID:

```
$ sudo cat /proc/mdstat
$ cat /proc/mdstat
Personalities : [raid5] [raid4]
md0 : active raid5 sdb1[0] sde1[3] sdd1[2] sdc1[1] 586075008 blocks level 5, 64k chunk,
algorithm 2 [4/4] [UUUU]
unused devices: <none>
```

Para ver más detalles sobre las unidades RAID:

```
$ sudo mdadm --query /dev/md0
/dev/md0: 558.92GiB raid5 4 devices, 0 spares. Use mdadm detail for more detail.
```

```
$ sudo mdadm --detail /dev/md0
/dev/md0:
Version : 00.90.03
Creation Time : Sat Jan 20 17:27:56 2007
Raid Level : raid5
Array Size : 586075008 (558.92 GiB 600.14 GB)
Device Size : 195358336 (186.31 GiB 200.05 GB)
Raid Devices : 4
Total Devices : 4
Preferred Minor : 0
Persistence : Superblock is persistent
```



Update Time : Sun Jan 21 22:23:05 2007

State : clean
Active Devices : 4
Working Devices : 4
Failed Devices : 0
Spare Devices : 0
Layout : leftsymmetric
Chunk Size : 64K
UUID : d65ce83c:150ba8ab:cfc213b0:81723f7b
Events : 0.3084

Number	Major	Minor	RaidDevice	State
0	8	17	0	active sync /dev/loop1
1	8	33	1	active sync /dev/loop2
2	8	49	2	active sync /dev/loop3

Para ver en detalle determinado dispositivo:

```
$ sudo mdadm --query /dev/loop1
```

Para examinar determinado dispositivo:

```
$ sudo mdadm --examine /dev/loop1
```

Este es un estudio inicial para solucionar la dificultad que se genera en el aprovechamiento de las computadoras de proyecto, ante la asignación estática de sus miembros. La solución propuesta está a prueba, y nuevos artículos del grupo de arquitectura pueden ser leídos al respecto. De idéntica forma debe escribirse la solución del home compartido.

Se recomienda implementar el sistema propuesto en los laboratorios de producción de la facultad y abrir el puerto 2049 que es el que usa NFS.



How To: configurar Fluxbox

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

Los sistemas GNU/Linux son ampliamente conocidos por su adaptabilidad a las necesidades de los usuarios. En este artículo se presenta un how-to para la configuración de Fluxbox, uno de los entornos de escritorio ligeros más usado.

Primero se realiza una instalación personalizada del sistema operativo, de modo que al terminar sólo quede la terminal. Una vez logueado como super usuario, se procede a instalar la interfaz gráfica:

```
[G/linuxinside]
# apt-get install xwindowssystem gdm fluxbox fluxconf idesk
```

Se inicia el servidor gráfico:

```
[G/linux-inside]
# /etc/init.d/gdm start
```

Se entra a la sesión, ahora iia configurar!!, pero primero se necesita un navegador de archivos y un editor de texto; es recomendable instalar desde este momento un Gestor de Paquetes (Aptitude, Synaptic, Adept, etc.):

```
[G/linux-inside]
# apt-get install synaptic pcmanfm pmount hal leafpad
```

Editar el fichero startup

Se edita el fichero startup, que se encuentra en el directorio .fluxbox del home del usuario para poner el fondo de pantalla y las aplicaciones que se ejecutarán el inicio de la sesión. Para poner el fondo de pantalla se utiliza la siguiente línea de comando en la consola:

```
fbsetbg /dir_de_la_imagen
```

Se listan las aplicaciones a ejecutar al inicio de la sesión en columnas dentro del fichero:

```
idesk &
gaim &
```

Se guardan los cambios y se reinicia Fluxbox (se hace click derecho en el escritorio y se selecciona Restart); se verán inmediatamente los cambios que se acaban de hacer.

Configurar el “look” del escritorio

Para configurar el tema de Fluxbox basta con dar clic derecho en el escritorio y entrar en el item “**Styles**” donde se escoge cualquiera de los temas disponibles; si los temas no satisfacen los gustos de cada cual siempre queda la opción de bajarse uno de internet, descomprimirlo y copiarlo en el directorio /usr/share/fluxbox/styles/.

Para lograr que las aplicaciones de Gtk usen el estilo y los iconos, que se desea, se crea un fichero con el nombre .gtkrc-2.0 en el home del usuario y se ponen las siguientes dos líneas:

```
Gtk-icon-theme-name="Tema_de_Iconos"
include "/usr/share/themes/Tema_de_Gtk/gtk2.0/gtkrc"
```

Ahora queda configurar **idesk** para definir iconos en el Escritorio. El primer paso es hacer una carpeta con el nombre .desktop en el home del usuario y ejecutar idesk & en la pseudo-terminal. **Idesk** creará por defecto el primer icono y el fichero de configuración .ideskrc en el home del usuario. Cada icono está definido por un archivo (con extension lnk) dentro del directorio .desktop, el cual tiene la siguiente estructura:

```
table Icon
```



```
Caption: Desktop
Command: pcmanfm ~/Desktop/

Icon: /home/user/images/icons/desktop.xpm
Width: 32
Height: 32
X: 30
Y: 30
end
```

El ítem **Caption** define el nombre del icono, **Command** el comando que se ejecuta, **Icon** el icono que mostrará, **Width y Height** el tamaño del icono y **X, Y** definen la posición del icono en el escritorio. Por defecto los iconos están configurados para ejecutar el comando asignado cuando tengan un doble-clic, esto se puede cambiar en el fichero `.desktop`. También se puede definir el tipo y color de fuente que usarán los iconos, sombra, transparencia, etc.

Teclas calientes

En el archivo `keys` que se encuentra dentro del directorio `.fluxbox` del home del usuario es donde se definen las teclas calientes, para hacer corresponder determinada acción con una combinación de teclas. Con el uso de la utilidad `fluxkeys` se podrán realizar estas configuraciones de forma rápida, sencilla e intuitiva.

Aplicaciones

Algunas aplicaciones alternativas a usar son **xpad**, utilidad para editar notas; **x SMB browser**, para navegar y gestionar recursos compartidos utilizando protocolo smb; **xpdf**, visor de archivos pdf; **pcmanfm**, navegador de archivos; **xscreensaver**, utilidad para los salva-pantallas; entre otros.

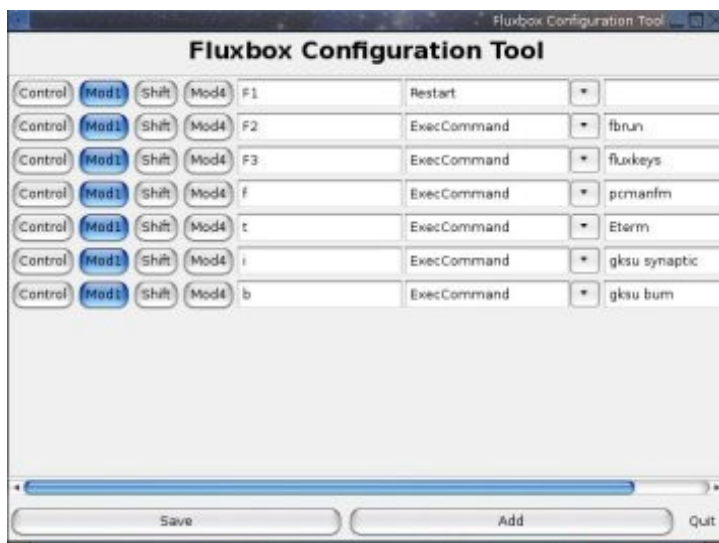


Figura 29: Herramienta de configuración de Fluxbox



David contra Goliat: migración de la MiniUCI de Granma

*Entrevista a Ing. Abel Meneses Abad
Por Msc. Graciela González Pérez
Noviembre de 2007*

Con laboriosidad de abejas, con solidaridad de hormigas, con un tesón increíble, un grupo de jóvenes cubanos de todas las edades pretende derrotar al software propietario personificado en el sistema operativo Windows, en una lucha sin cuartel.

El capitán de estos Quijotes modernos es tan joven que puede confundirse con sus estudiantes; adarga al brazo, todos están decididos a poner su grano de arena en esta batalla contra el gigante. Así, durante 3 semanas, desde el 17 de Septiembre al 9 de Octubre, el Ing. Abel Meneses Abad, miembro del Grupo Técnico Nacional de SWL en Cuba y líder del proceso de migración a software libre de la facultad X, apoyó junto a miembros del proyecto Servicios y Soporte para la Migración a SWL (UNICORNIO) la migración de la Facultad Regional de la UCI en el Oriente de Cuba.

Aunque la curiosidad mató al gato, quisimos dejar constancia en esta entrevista que se realizó virtualmente y que hoy compartimos con nuestros lectores; de los pormenores de la tarea.

Abel, ¿cuál fue la tarea que los llevó a la mini UCI de Granma?

El comienzo de las facultades territoriales marcó un momento importante para el desarrollo regional de la informática en Cuba. Durante 3 años se había hablado de la importancia de migrar la UCI a SWL. Dos de los principales cuadros de la facultad 10, que desde hacía dos años usaba SWL, fueron enviados a Ciego de Ávila y Granma.

En Granma, Pedro Álvarez Barreras, vicedecano de formación, sugirió que la mini UCI de esa provincia comenzara a funcionar usando software libre. Las excelentes relaciones de los antiguos directivos de la facultad 10 con la UCI, y el trabajo conjunto (o colaborativo) hicieron posible que fluyera la intención de hacer realidad una facultad con tecnologías únicamente libres.

Por ello se concretó que a principios del curso 2007 - 2008, y tras la consulta con la máxima dirección de la universidad, se comenzara el año lectivo en septiembre del 2007 utilizando tecnologías de código abierto. El 16 de noviembre viajé a Bayamo, donde me esperaban Alexander Martínez Fajardo y Frank Rosales Muñoz, estudiantes de la UCI que pertenecen al Proyecto UNICORNIO. Tres días más tarde viajamos a Manzanillo, donde se encuentra ubicada la mini UCI, para comenzar el proceso de migración.

¿Cuáles eran los objetivos a corto y mediano plazo con respecto a la tarea?

El objetivo era conseguir que todos los procesos de la facultad regional pudieran hacerse con SWL: la docencia, el trabajo de las secretarías, los diferentes departamentos, laboratorios, aulas, etc. A largo plazo, debíamos potenciar el trabajo con software libre de toda la región oriental, y comenzar a crear un foco grande de ingenieros que conocieran, un polo científico donde se hablara de software libre. Luego, promover encuentros con empresas, organismos y direcciones regionales.

A mediano plazo pretendemos impregnar a la universalización de la enseñanza en los municipios con las tecnologías libres. Promover comunidades como las de Buey Arriba y Puerto Padre en otros municipios de la región, que permitan el desarrollo de los procesos de informatización en los lugares más intrincados. También la organización de eventos y centros de producción para la región, el país y tal vez para la exportación de productos internacionales de software.

¿Qué encontraron allá y cómo desarrollaron la tarea?

En Granma encontramos la idiosincrasia de la región. La amabilidad característica del pueblo que inició las luchas por la independencia. El calor de la gente, la afabilidad. Y la convicción, además de una excelente organización para alcanzar el objetivo que ya expliqué.



Estuvimos dos semanas haciendo un trabajo intenso; luego fue más pausado, y después de esos días los estudiantes que trabajaban conmigo regresaron y solo quedé yo con el grupo de soporte de la mini UCI de Granma. Fue necesario pasar imagen en los laboratorios, bajar un repositorio completo de Fedora 7, probar las aplicaciones que se usarían en la docencia, redactar la documentación que dejaríamos a los compañeros de allá, entre otras cuestiones. Tuvimos varias reuniones con Pedro, la decana Reyna Estrada y otros dirigentes, y sostuvimos un fructífero encuentro con los compañeros de las FAR en el territorio.

¿Cómo es la mini UCI, y cómo es la gente por allá?

La mini UCI es un lugar bastante interesante. Por un lado toda la estructura comprende además un politécnico de informática donde hay jóvenes de entre 15 y 18 años de edad, todos de la provincia de Granma. Y están los jóvenes del primer año de la UCI, nuestros recién graduados con algunos egresados de la Universidad de Oriente y profesores experimentados provenientes de la propia Universidad de Granma.

Toda esta heterogeneidad permite que exista una red de relaciones humanas diferentes a las que comúnmente encontramos en las universidades. Y por lo tanto se produce un intercambio muy rico en el que, a mi entender, todos salen favorecidos. Nuestra gente -pues yo soy bayamés- es muy humana. Y eso siempre es visible. Es una experiencia sin igual; allí no hay pase para los estudiantes, en ese sentido es como otras universidades, y diferente a la UCI. Aunque un poco retirados, a menos de 1 km (distancia que caminamos en la UCI todos los días), están el mar, policlínicas, un malecón y gran parte de la ciudad. También el famoso Hospital Celia Sánchez de quemados y la escuela de medicina de Granma.

¿Alguna anécdota interesante que contar?

Anécdotas hay muchas, personales y relacionadas con el software libre. Pero me referiré a las segundas. No habíamos tenido la oportunidad en años en la UCI de compartir con los administradores del nodo, fue muy importante hacerlo en Granma, pues esto ayudó a realizar las tareas de migración mucho más rápido. Compartimos con los profesores de Ajedrez, a los cuales mostramos Infodrez (la aplicación Web para el desarrollo de juego online), y quedaron sorprendidos al descubrir el software libre, y algunos productos disponibles en Internet.

El brillo de unos ojos que descubren un mundo aparentemente negado, fue lo mejor que he visto en años, durante la lucha por la implantación del software libre en Cuba. Un día estuvimos 3 horas para configurar la impresora de la decana. Dijimos: "aquí mismo terminó el compromiso de la dirección con el SWL", pero nada de eso. Al día siguiente en la mañana, Yoel, el administrador de la mini UCI, hizo dos o tres pasos que ya habíamos repetido, ¡y listo!, impresora imprimiendo. Terminaron nuestras angustias.

Pasamos momentos sin red, sin luz eléctrica. Dimos conferencias a estudiantes y profesores. Probamos nuestros softwares y métodos de pasar imagen. Cada día fue importante en el intercambio con profesores y estudiantes, en la beca, en la calle, para convencer a todos de que era el mejor camino para nuestro país y para los futuros ingenieros.

¿Consideras que se cumplió el objetivo?

El objetivo fue cumplido: tras dos meses de prueba, Granma no solo mantiene todas sus tecnologías con software libre (exceptuando capital humano, contabilidad, la transmisión de televisión y Akademos), sino que aporta a la comunidad UCI algunas experiencias valiosas que también estamos aplicando aquí, en la sede central.

¿Qué expectativas te quedaron para el futuro?

Nos queda el evento regional de Granma, que será a finales de noviembre de 2007. Y pensamos estar a principios del próximo curso, cuando la sede sea en Bayamo, para impulsar a nuestros compañeros a metas superiores y entregarles nuestros avances, además de compartir los de ellos. Tenemos pensado poner en práctica la misma experiencia en la facultad regional de Ciego de Ávila.

Desde esta redacción, nos comprometemos a llevarle las primicias de esta cruzada en la que David le va ganando cada vez más espacios a Goliat.



Figura 30: El Rector de la UCI, Ing. Melchor Gil Morell, en discurso pronunciado en la mini UCI de Granma



Figura 31: El entrevistado, Ing. Abel Meneses Abad



Cómo crear y usar una librería estática

Jorge Luis Betancourt González
jlbetancourt@estudiantes.uci.cu
Proyecto FILPACON

Resumen

En el mundo de la programación es muy común utilizar librerías²⁸ de terceras personas, pero no lo es tanto encontrar a programadores, exceptuando los muy experimentados, que escriban librerías para sus propios proyectos. En este artículo se trata de mostrar cómo crear una librería utilizando el entorno de programación Code::Blocks. Lo que se expondrá está pensado para ser utilizado en ese entorno de desarrollo, pero sirve de base para adaptar a cualquier otro entorno, como pueden ser Anjuta y KDevelop, por sólo citar 2 de ellos.

Introducción

Como programador se debe ser una persona ordenada, al menos en lo referente a programas. Esto puede no ser del todo cierto, pero debería. Igual de acertado es que gusta de realizarse la misma tarea cierto número de veces, lo cual incluye escribir varias veces el mismo código en diferentes partes de un proyecto, o proyectos distintos; es por ello que se busca que el ordenador haga por uno la mayor parte del trabajo.

Debido a esta problemática surgen las librerías, las cuales básicamente se tratan de colecciones de funciones y tipos de datos que necesitamos usar frecuentemente. Las ventajas de las librerías son varias y una de las más importantes es el hecho de que permite mantener cierta modularidad sobre la aplicación que estemos desarrollando.

Lenguajes como C y C++ incluyen la posibilidad de reutilizar el código mediante librerías, ahorrando muchas horas de trabajo. Es muy común cuando se comienza a programar en C y C++ que se pregunte si no existe una forma más sencilla de reutilizar el código que ya se ha escrito en proyectos anteriores. La respuesta a esta interrogante es sí: se puede escribir nuestras propias librerías e incluirlas en nuevos proyectos.

En este artículo se explica cómo crear librerías estáticas²⁹, del mismo tipo de las que incluye cualquier compilador como librerías ANSI. Así, cada uno podrá crear sus propias librerías, que contengan las funciones que use más frecuentemente, de una forma ordenada y sobre todo, siempre accesibles, sin necesidad de escribir el código nuevamente.

Un ejemplo

Este artículo se centrará en un IDE (Entorno de Desarrollo Integrado, por sus siglas en inglés) concreto: Code::Blocks, de modo que lo que se describa aquí no será del todo general, pero sí deberá constituir una base para poder adaptarse a cualquier entorno de programación que se utilice sin demasiados cambios. A la hora de crear las librerías no se debe mezclar funciones y tipos de datos sobre diferentes temas; sería incorrecto escribir una librería que contenga funciones para el tratamiento de cadenas (strings), para la resolución de ecuaciones diferenciales y además para el tratamiento de imágenes. Es más viable, tanto para el usuario final que utilizará la librería como para los desarrolladores y actualizadores, crear librerías separadas para cada tema.

²⁸ Una librería es un conjunto de tipos de datos y funciones que permiten modularizar y reutilizar código de forma versátil en proyectos. Por convenio, los nombres de todas las librerías estáticas comienzan por lib y tienen .a por extensión.

²⁹ Una librería estática es una colección de tipos de datos y funciones que son incluidas en el binario de la aplicación en tiempo de ejecución.



Las librerías deben estar bien documentadas, para facilitarle al programador que las utilice la tarea de documentarse sobre lo que realiza cada función. Para la gestión de la documentación de cada una, se recomienda la utilización de doxygen³⁰.

Para ilustrar el proceso de creación de una librería estática en Code::Blocks se creará una librería con funciones para manejar cadenas, que complemente a la librería estándar de C diseñada para tales fines. Se incluirán funciones para convertir cadenas a mayúsculas y a minúsculas, para comparar cadenas sin importar el tipo de carácter (sin distinción de mayúsculas o minúsculas). Por supuesto, se debe tener presente que la utilidad del código expuesto en este artículo es meramente didáctica, y por tanto es poco probable que sea utilizable en alguna aplicación real.

En Code::Blocks, lo primero que hay que hacer es crear un nuevo proyecto de tipo “Static Library” (Librería estática), que se encuentra en Projects / Console / Static Library (Figura 32). Seleccionado este ítem, lo próximo que se verá es el asistente para la Creación de Librerías Estáticas (Figura 33). El nombre que del proyecto es importante, como se verá más adelante, ya que es el nombre que tendrá el fichero objeto generado que contendrá los binarios con la librería. A este proyecto se le llamará “libmystrings”.

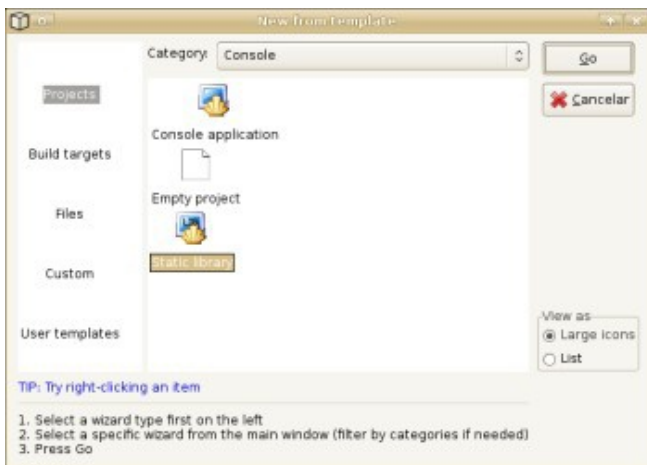


Figura 32: Creación de un nuevo proyecto en Code::Blocks



Figura 33: Asistente para creación de una librería estática

Lo segundo, y este paso es muy importante, es crear un fichero de cabecera con los prototipos de las funciones y los tipos de datos que incluirá la librería. Realmente bastará con incluir los tipos de datos y los prototipos de las funciones que se desean compartir (hacer públicas), es posible que se tengan funciones privadas que no tienen por qué aparecer en el fichero de cabecera. Esto, por supuesto, trae como consecuencia que los tipos de datos y las funciones que no aparezcan en el fichero de cabecera se podrán utilizar solamente en el interior de la librería, pero no fuera de la misma. Por defecto, el Code::Blocks crea un fichero main.c que contiene un prototipo de una librería estática, se puede optar por editar este fichero o eliminarlo del proyecto y crear uno nuevo que sería finalmente el fichero de cabecera para la librería.

Por lo tanto, se añade un fichero al proyecto, al que se llamará “mystrings.h”. Este fichero será el que se incluya en la zona de cabeceras en los programas que usen funciones de esta librería. A continuación se presenta el código fuente de dicho fichero:

```
/*  
* mystrings.h
```

³⁰ doxygen es un generador de documentación para código fuente con soporte para varios lenguajes como: C++, C, Java, Objective-C, Python, etc., que funciona en la mayoría de Sistemas Unix, Windows y Mac OS X. doxygen es un acrónimo de dox (Document) gen (Generator), escrito mayormente por Dimitri van Hessch, proyectos como KDE lo utilizan para generar la documentación de sus API's.



```
*
* Fichero de cabecera para la librería libmystrings.h
**
* Convierte la cadena s1 a mayúsculas
*/
char *strToMay(char *s1);
**
* Convierte la cadena s1 a minúsculas
*/
char *strToMin(char *s1);
```

El siguiente punto es implementar las funciones de la librería. Para ello se añade un nuevo fichero al proyecto, en este caso llamado “mystrings.cpp”:

```
/*
* mystrings.cpp
*
* Fichero de implementación para la librería libmystrings
*/
#include <cctype>
#include <iostream>
#include <cstring>

char *strToMay(char *s1)
{
    char *str = s1;
    while(str && *str) {
        *str = toupper(*str);
        str++;
    }
    return s1;
}

char *strToMin(char *s1)
{
    char *str = s1;
    while(str && *str) {
        *str = tolower(*str);
        str++;
    }
    return s1;
}
```

El siguiente paso es compilar el proyecto. Evidentemente, no hay nada que ejecutar. El resultado de compilar una librería estática en Code::Blocks es un fichero con la extensión “.o”, en este ejemplo se obtiene un fichero con el nombre “libmystrings.o”.

Usar una librería

Ya se tiene una librería funcional pero entonces es muy probable que surja la pregunta: ¿cómo usar esa librería? La respuesta es muy sencilla: Se crea otro proyecto para probar la librería, esta vez será un ejecutable de consola, al que se le llamará “usolibmystrings”. Se añade un fichero al proyecto, “usolibmystrings.cpp”, con el siguiente código:



```
/*  
* mystrings.cpp  
*  
* Fichero de implementación para la librería libmystrings  
*/  
#include <cctype>  
#include <iostream>  
#include <cstring>  
  
char *strToMay(char *s1)  
{  
    char *str = s1;  
    while(str && *str) {  
        *str = toupper(*str);  
        str++;  
    }  
    return s1;  
}  
  
char *strToMin(char *s1)  
{  
    char *str = s1;  
    while(str && *str) {  
        *str = tolower(*str);  
        str++;  
    }  
    return s1;  
}
```

Como se puede observar, en el código fuente se debe adicionar una sentencia 'include' para indicarle al compilador dónde se encuentra el fichero de cabecera, y que pueda realizar la comprobación de tipo. Es por ello que es necesario distribuir el fichero de cabecera conjuntamente con el binario de la librería.

Otro paso importante es incluir la librería en la fase de enlazado. Para ello se acude a “Las opciones de enlazado” (Project / Properties / Project's Build Options / Debug (Release) / Linker Settings /) Luego en la sección “Librerías a enlazar” (Link libraries) se presiona en Añadir (Add) y se adiciona el fichero “libmystrings.o” (Figura 34). Luego de esto bastaría con Compilar el proyecto (Ctrl + F9) y ejecutarlo, el resultado podrá ser observado en pantalla, tal y como muestra la Figura 35.



Figura 34: Opciones de enlazado de la aplicación donde se incluye la nueva librería para ser enlazada en tiempo de compilación.

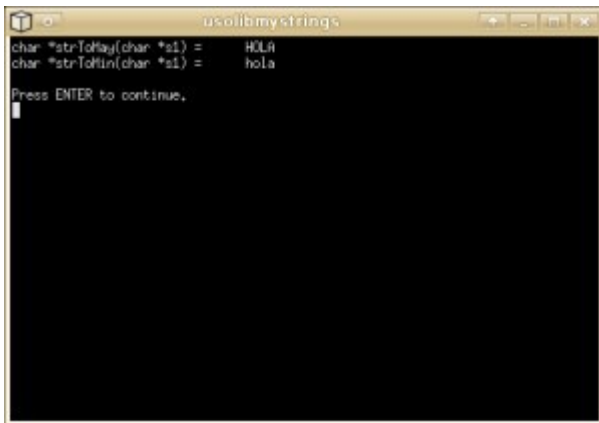


Figura 35: Resultado de la ejecución de la aplicación libmystrings.

Conclusiones

Es evidente la importancia que reviste la creación, por parte de los programadores, de una librería para proyectos propios. Este artículo constituye una base para que los programadores noveles, y por qué no, también para que los menos noveles que lo precisen, puedan escribir sus propias librerías, lo que a su vez les permitirá hacer más modulares sus aplicaciones, aislar mejor los errores y compartir sus librerías con el resto de la comunidad.



Creando un tema de boot splash

Yunier Vega Rodríguez, Eiger Mora Moredo
yvrodriguez@estudiantes.uci.cu, emora@estudiantes.uci.cu
Proyecto UNICORNIOS

Un artículo publicado anteriormente en la revista UXi trataba sobre la creación de un tema de GDM y la importancia que tenía la concepción de su diseño más que la programación implícita en el hecho de cambiar la estructura de una interfaz en particular. En esta ocasión toca un tema no menos interesante: la creación de temas de boot splash en los sistemas GNU/Linux.

¿Qué es boot splash?

Del inglés *boot* (arrancar) y *splash* (salpicadura), el boot splash vendría siendo, en un acercamiento a nuestro idioma, la “presentación de inicio del sistema”, caracterizada por imágenes de fondo (*background*) y una barra de progreso para indicarle al usuario la evolución de la activación de los programas y servicios fundamentales en su ordenador cuando el sistema operativo se inicia. Además de estos componentes, el boot splash contiene animaciones realizables en aplicaciones de diseño gráfico como Gimp, tema al cual nos referiremos en próximas ediciones.

Estructura

Localizado en `/etc/boot splash/themes/debian`, un tema de boot splash es un directorio que contiene a su vez tres subdirectorios, nombrados comúnmente animaciones, config e images donde se encuentran distribuidos sus componentes, que análogamente a como ocurría en la elaboración de un tema de GDM a través de la edición de un fichero con extensión xml, su gestión corre a cargo de `boot splash-1024x768.cfg` situado en config, y cuya estructura se muestra en la Figura 1.

```
# name of the picture file (full path recommended)
jpeg=/etc/boot splash/themes/tocororo-desktop/images/boot splash-1024x768.jpg
silentjpeg=/etc/boot splash/themes/tocororo-desktop/images/silent-1024x768.jpg

progress_enable=0

# (ax, ay) are the (x, y) coordinates for playing boot/shutdown animations
ax=375
ay=145

# background
box silent noover 640 431 890 437 #04045498
box silent inter 640 432 640 436 #b6e3f0 #b6e3f0 #b6e3f0 #b6e3f0
box silent 640 432 890 436 #b6e3f0 #b6e3f0 #b6e3f0 #b6e3f0

# black border
box silent 640 431 890 431 #313234
box silent 640 437 890 437 #313234
box silent 640 431 640 437 #313234
box silent 890 431 890 437 #313234
```

Figura 1: La imagen muestra una parte del contenido del fichero
`boot splash-1024x768.cfg` del directorio config



Imágenes boot splash y silent

Encerradas en el recuadro de color verde de la figura, las líneas de código muestran la dirección de las imágenes boot splash y silent (imágenes background), que se visualizan en el inicio del sistema (la primera) y como background del modo consola de éste (la segunda), asignadas a las variables jpeg y silentjpeg que permiten reconocerlas.

Estas imágenes se sitúan en el subdirectorio images como puede apreciarse, y debido a que otra localización no es permitida, en nuestro proceso de creación las habremos de colocar, después de diseñadas, en este mismo sitio.

Lo interesante...

Sin dudas lo más interesante del proceso de creación de un tema de boot splash es la barra de progreso, en la que más que su forma (la cual es rectangular) interesan, a los efectos de la configuración del fichero .cfg, sus colores y localización en la pantalla. En la figura mostrada aparecen encerradas en un recuadro rojo las líneas de código que definen casi explícitamente los bordes de la barra (código encerrado en azul) y su fondo (código encerrado en amarillo).

Una interpretación de dichas líneas se ofrece a continuación para que el usuario interesado pueda modificarlas a gusto, conforme a sus preferencias.

Área y bordes de la barra de progreso

En un análisis del recuadro con bordes amarillos, la línea precedida por la sentencia *box silent noover* identifica el fondo base de la barra, en tanto las sentencias *box silent inter* y *box silent* intervienen en el diseño de la animación: la primera define los colores de la parte superior de la barra animada, en tanto la segunda la inferior. Los valores, tal y como se nos presentan representan lo siguiente:

box silent "algo" <inicio en x ><inicio en y><final en x ><final en y>

En el caso de los colores, los mismos son asignados a las posiciones indicadas, de manera similar. El caso anterior es extensible también a los bordes de la barra de progreso, cuyo código aparece señalado en el recuadro azul. Como puede apreciarse en este caso, pueden definirse tantas líneas como se quiera con el color deseado, tal cual en la barra de progreso de la Distribución Ubuntu, de moda por estos tiempos.

Activar el tema de boot splash

Una vez concluida la etapa de diseño del boot splash, activarlo es una tarea bastante sencilla, basta con crear en el directorio /etc/boot splash/themes/debian un enlace simbólico denominado current a la carpeta donde se encuentra nuestro tema. Finalmente se ejecuta en la consola **update-initramfs -u** y listo.

Arranques en Windows y GNU/Linux

Nada hay más tedioso para la generalidad de los usuarios que tener siempre la misma configuración gráfica de sus sistemas, y GNU/Linux ofrece la ventaja de poder cambiarla uno mismo, sin la necesidad de que otros decidan lo que debemos o no ver, algo que se aplica sobre todo a los sistemas operativos privativos como Windows, cuyas interfaces son esquemáticas y obedecen a los patrones definidos por su creador, no a los gustos y deseos del usuario final, siempre para ellos relegados a un segundo plano. Con el texto de este artículo hemos querido, de una manera sencilla, demostrar una vez más la flexibilidad de los sistemas libres en la configuración de sus entornos gráficos, esta vez en el propio arranque del Sistema.



Cambiar apariencia de servidor Apache

Nelio Veliz Pedraza
nveliz@estudiantes.uci.cu
Administrador del Servidor de Documentación
Proyecto UNICORNIO

La apariencia que trae la interfaz del servidor Apache ha sido criticada por mucho tiempo, pues no es la más agradable para los usuarios; por ello se ha creado con PHP una herramienta que permite cambiar la apariencia del mismo, siendo esta configurable según el gusto de la persona.

Index of /Documentacion

Name	Last modified	Size	Description
Parent Directory		-	
Ciberderecho/	19-Mar-2007 17:56	-	
Design/	20-Oct-2007 15:40	-	
Distribuciones/	01-Mar-2007 00:00	-	
Docencia/	20-Oct-2007 16:11	-	
Documentacion para la Casa.tar.gz	26-Jun-2007 03:38	483M	
Generales/	26-Oct-2007 23:35	-	
Literatura/	20-Oct-2007 16:17	-	
Migracion/	04-Feb-2007 18:58	-	
Programacion/	20-Oct-2007 16:34	-	
Publicaciones/	27-Jun-2007 22:08	-	

Apache/2.2.3 (Debian) DAV/2 SVN/1.4.2 PHP/4.4.4-8+etch3 Server :

Figura 36: Apariencia por defecto de la interfaz del servidor Apache.

El programa que permite el cambio de apariencia le interfaz del servidor Apache fue creado por Humberto Salazar Pacios. Conjuntamente con la comunidad de software libre, los autores del presente trabajo le han realizado algunas modificaciones. Dicho programa posee una carpeta nombrada "FileSystemCore" que contiene todos los archivos que pueden ser modificados, la cual debe colocarse en /var/www.

El directorio "fileType" donde se encuentran los iconos que saldrán en el servidor los cuales pueden ser cambiados siempre y cuando tengan extensión .jpg (se sugiere que sean de tamaño 22*22). Para que la apariencia cambie recursivamente en cada directorio del árbol tiene que existir el archivo "index.php" por lo que la comunidad ha desarrollado un script el cual se encarga de colocarlo recursivamente, para su buen funcionamiento no pueden existir directorios con espacios ni archivos con puntos aparte del de la extensión.

Por ejemplo:

documentacion.2007.doc Ejemplo incorrecto
documentacion_2007.doc Ejemplo correcto

La herramienta pondrá por defecto el siguiente icono: en caso de encontrar algunos de los errores antes mencionados.



Figura 37: Icono que aparece por defecto en caso de errores.



Es importante destacar que esto no significa que se pierda el contenido del mismo, al ser arreglado el nombre pondrá el icono que le corresponda. A continuación se muestra el código del script mencionado:

```
#!/bin/bash
#Script para poner recursivamente index.php en cada directorio.
#Comunidad SoftwareLibre
#Directorio al cual se quiere hacer la función.

cd /mnt/datos/Documentacion
func_rekursiva()
{
for i in *
do
(
if [ -d "${i}" ]
then
cd "${i}"
#dirección donde se encuentra el archivo
cp /mnt/datos/Documentacion/index.php
$(pwd)/func_rekursiva
if
)
done
}
func_rekursiva
#En caso de existir otro directorio que no esté en el árbol anterior simplemente ponemos:
cd /mnt/datos/Media
func_rekursiva
```

Para el buen funcionamiento de la herramienta se necesita tener instalado php4, no obstante pueden modificarse ciertos archivos y hacer que funcione también con php5. Como resultado la nueva interfaz del servidor Apache quedaría como se muestra en la figura:



Figura 38: Nueva apariencia del servidor Apache.



El Software Libre y los procesos pedagógicos contemporáneos

Orlando Cárdenas Fernández
gabriel@uci.cu
Universidad de las Ciencias Informáticas

Resumen

El Software Libre, en tanto forma de organización de la ciencia y la tecnología de profundas implicaciones sociales, está llamado a incidir notablemente en los procesos pedagógicos contemporáneos, tales como la universalización del conocimiento, inclusión social, formación de valores, innovación, formación masiva del capital humano, internacionalización e integración regional universitaria, virtualización y semipresencialidad, entre otros.

En contraposición al software privativo, las plataformas libres, por su esencia, están sustentadas en principios éticos solidarios y colectivistas, constituyendo una alternativa socializadora y antimonopolista para los procesos educativos de la enseñanza superior del siglo XXI. La integración eficiente de la formación profesional con la producción y la investigación científica, asumida en el contexto de las funciones sustantivas de la Educación Superior contemporánea, alcanza un desarrollo más coherente cuando descansa en las posibilidades altruistas del Software Libre.

Desarrollo

Los programas informáticos pueden ser clasificados por su concepción social en:

Privativos: su propiedad absoluta permanece en manos de quien posee sus derechos y no del usuario, quien solo puede usarlo bajo ciertas condiciones de licenciamiento. Se priva al usuario del uso, modificación y redistribución. Da derechos a usarlo solamente “tal y como es.”

Libres: dan posibilidades de usarlos, estudiarlos, modificarlos, mejorarlos, adaptarlos y redistribuirlos con la única condición de no agregar ninguna restricción adicional al software modificado. Deben permitir el acceso al código fuente como requerimiento “sine quanon” para ejercer los derechos planteados. Pueden ser gratis o comerciales.

El Software Libre garantiza los siguientes derechos:

1. “Usarlo para cualquier propósito”, lo podemos utilizar con fines educativo, cultural, comercial, político, etc.
2. “Estudiar su funcionamiento y adaptarlo a las necesidades” Podemos estudiar su funcionamiento que nos permite descubrir funciones ocultas, otras posibilidades, qué le falta para realizar alguna acción. Adaptarlo a las necesidades implica poder suprimirle partes que no interesan, agregarle otras, etc.
3. “Distribuir copias”. Somos libres de redistribuir el programa, ya sea gratis o con algún precio, por correo electrónico, FTP o en CD, a una persona o a varias.
4. “Mejorar el programa y liberar las mejoras al público”. Tenemos la libertad de mejorar el programa, o sea hacer menores los requerimientos de hardware para funcionar, mayores prestaciones, que ocupe menos espacio, etc. Poder liberar las mejoras quiere decir que somos libres de poder redistribuir ese programa mejorado o simplemente proponer las mejoras en un lugar público (foro de noticias, lista de correos, sitio Web, FTP, canal de Chat)

La contribución más importante del Software Libre a las tecnologías digitales es que facilita copiar y modificar la información, divulgar y mejorar el conocimiento. La aparición de la imprenta popularizó el conocimiento y creó una industria poderosa, donde acceder a los conocimientos se convertiría en una ventaja importante sobre los demás, privatizándose el acceso discriminatoriamente. La industria de distribución de la información se basó en productos físicos “escasos” con elevados costos de reproducción y distribución. Ahora los costos son mucho menores. ¿Es moralmente aceptable la exclusión social?



El software está llamado a ser el nuevo depositario del conocimiento humano. La comunidad del Software Libre esta comprometida con la idea acerca de que los elementos ejecutables de la tecnologías pueden ser producidas sin relaciones de propiedad discriminatorias.

El Software Libre es ante todo una actitud ética, política y social en tanto nueva forma de organización de la ciencia y la tecnología en su estrecha relación con el desarrollo social. Resulta muy sorprendente escuchar al Vicepresidente de Microsoft argumentar que el Software Libre contradice el “sueño americano”, cuando es justamente el software privativo quien se da de narices contra la ilusoria doctrina del pragmatismo estadounidense, que asegura igualdad de oportunidades para todos de escoger su propio camino y además tener éxitos si trabaja con tesón.

Las restricciones que establecen las licencias de software privativo niegan la igualdad de posibilidades para el desarrollo profesional. Las restricciones que establecen las licencias de software privativo niegan la igualdad de posibilidades para el desarrollo profesional. El software ha pasado a ser considerado casi como un producto físico comercializable, pero sólo en ciertos sentidos, porque sus productores han impuesto aun mayores restricciones (prestar un libro es legal, pero prestar un programa a un amigo viola los términos de las licencias de la propiedad intelectual).

Las restricciones impuestas al software para que pueda ser comercializado, son tan artificiales que tuvieron que ser sustentadas mediante leyes restrictivas, si nos atenemos a las de propiedad intelectual y patentes casi el 100% de los usuarios de ordenadores personales está cometiendo un delito, a menos que use exclusivamente Software Libre.

El Software Libre por su naturaleza estimula la innovación tecnológica, masifica la producción de programas, crea programadores mas eficientes y pertrechados con una ética mas humana. Tiene valores concordantes con el concepto de globalización humanizadora y solidaria. El mercado informático tiende al monopolio de productos en todos sus ámbitos. Los usuarios quieren rentabilizar el esfuerzo realizado en aprender como funciona un programa, las empresas quieren encontrar gente formada en el uso de sus softwares y todos quieren que los datos que procesan puedan ser entendidos por los programas de las empresas y las personas con las que se relacionan.

Cualquier iniciativa que intente romper una situación donde el producto domine claramente al mundo, está destinada a producir mas de lo mismo, si tiene éxito vendrá otro producto a ocupar ese nicho y un nuevo monopolio. Sólo los cambios tecnológicos pueden producir durante un tiempo, la inestabilidad suficiente como para que nadie domine claramente. Pero la situación donde un producto domina el mercado no es necesariamente indeseable, lo preocupante es cuando hay una empresa que lo controle.

En el mundo del software privativo un monopolio de producto en un segmento cualquiera equivale al monopolio por parte de la empresa que lo produce. Sin embargo en el caso del Software Libre un monopolio de producto no se traduce automáticamente en un monopolio de empresa. Si el producto es libre, cualquier empresa puede trabajar con él, mejorarlo, adaptarlo a las necesidades del cliente y en general, ayudar en su evolución. **En el Software Libre puede existir monopolio de productos pero no monopolio de empresas.**

El Software Libre ataca el sustento filosófico y ético de la propiedad capitalista del conocimiento y no solo en la teoría sino en la práctica, porque los programas diseñados en esta plataforma están a disposición de todos los usuarios, sin costo de licencia porque no pertenecen a ninguna empresa y tienen la misma o mejor calidad funcional que los programas privativos. Estos programas con sus códigos libres y abiertos son generados por cientos de miles de programadores voluntarios, que trabajan en casi todos los países y coordinan sus tareas a través de Internet.

Cuando analizamos el fenómeno tecnológico del Software Libre y nos atenemos a su impacto y significación social, estamos en el deber de destacar ante todo su connotación ética, en tanto alternativa socializadora y antimonopolista al software privativo. La moral actúa como el sistema de normas, principios y valores que



regulan las conductas humanas de una determinada sociedad, expresión de su noción del bien y el mal, el deber y el honor, en un contexto histórico definido.

Los valores guían la actividad humana en todas sus manifestaciones, convirtiéndose en elementos indispensables en la regulación de la conducta de los seres humanos. Son contenidos sociales ponderados, defendidos, apreciados, buscados y considerados importantes por toda la sociedad o un grupo, actuando como reguladores del comportamiento del sujeto. Los valores morales se presentan como acciones, conductas, principios, ideales y sentimientos que posean una significación regulativa y determinante en la vida colectiva. Cada profesión engendra una determinada ética que denominamos ética profesional, en ese sentido la ciencia que estudia las normas y valores morales de la informática la conceptualizamos como Ética Informática.

El Software Libre es un sistema de trabajo totalmente voluntario y cooperativo, los líderes de cada proyecto son elegidos democráticamente, de acuerdo a sus méritos, por los propios integrantes. Este modelo de trabajo elimina la clásica alienación del trabajador en el sistema capitalista, porque todos están directamente interesados en el producto final, el trabajo es voluntario y no forzado, el producto final les pertenece y además lo donan para toda la humanidad. La actitud altruista y humanista de estos programadores libres que priorizan el reconocimiento moral e intelectual de su trabajo antes que el económico, es digna de admiración e imitación, porque no reciben pago por concepto de licencia, aunque reciben, como es justo, retribuciones por su trabajo. En realidad, se parece a una gran cooperativa mundial, en la cual todos "depositan" sus conocimientos en forma de programas en un gran repositorio y retiran lo que necesitan para ellos mismos o para dar servicios a sus clientes.

Esta filosofía del Software Libre ataca al Monopolio Tecnológico Capitalista porque:

- El autor es colectivo.
- La obra se perfecciona continuamente en todas partes por medio de otros desarrolladores.
- Está a disposición de todos los usuarios.
- Le proporciona a todos la oportunidad de trabajar en el desarrollo y servicio de programas.

¿Qué valores morales sustentan la ética del Software Libre?

1. Solidaridad
2. Colectivismo
3. Honestidad
4. Responsabilidad social
5. Sensibilidad humana
6. Altruismo

Estos valores se expresan en su connotación e impacto sobre el desarrollo social. El Software Libre garantiza la soberanía tecnológica de los países, fundamentalmente de los subdesarrollados, favoreciendo el proceso de identidad nacional (idiomas nacionales, culturas autóctonas); a diferencia del software privativo que profundiza y fortalece la transculturación globalizante y conlleva a la pérdida de autonomía. Este movimiento puede llegar a convertirse en una forma de lucha de clases contra la propiedad privada en el entorno digital, una forma de luchar contra el imperialismo y su dominación tecnológica planetaria.

El Software Libre está llamado a reducir la brecha digital entre inforricos e infopobres, en primer lugar por la libertad que otorga de modificar y redistribuir copias mejoradas públicamente, de hecho mas baratas que el software privativo y muchas veces gratuitas.

La formación masiva de recursos humanos para los países del tercer mundo, es una ventaja que trae esta forma de organización del desarrollo tecnológico, verificada a partir de la multiplicación de desarrolladores de software, a diferencia del software privativo que multiplica los piratas y reduce los desarrolladores. La piratería del software privativo incide negativamente en el desarrollo del Software Libre por el desperdicio de talentos tecnológicos que se enfrasan en esa actividad, pudiendo ser utilizados en el desarrollo de programas informáticos sobre plataformas libres.



Si el software privativo favorece los intereses de las multinacionales de la información, el Software Libre propicia el desarrollo de la pequeña y mediana empresa productora de servicios informáticos, y en ese sentido es una actividad estratégica para el progreso de los países subdesarrollados.

No por casualidad los líderes de los monopolios internacionales de la informática se muestran tan preocupados por el avance del Software Libre. El Vicepresidente para América Latina de la multinacional estadounidense Microsoft Eugenio Beaufrand, planteó que el uso del Software Libre es perjudicial para el desarrollo de la región, por el impacto negativo que tiene sobre los programadores y se refirió a casos como Uruguay que exporta software por 100 millones de dólares al año. Obviando lógicamente la cifra que tiene que pagar este país por concepto de licencias de software privativo.

El directivo se sintió preocupado por el auge de los gobiernos de estimular el Software Libre como Brasil y Venezuela. Microsoft obtiene 800 millones de dólares en Latinoamérica (4 % de su negocio mundial), donde Brasil y México son sus principales mercados, acaparando el 60 % de los ingresos de la región. El Software Libre esta llamado a convertirse en un factor potencial de integración latinoamericana en los marcos del nuevo escenario social y político de la región, donde la interacción ciencia, tecnología y desarrollo social constituye un elemento importante del proceso global que se desarrolla en las diferentes alternativas integracionistas, fundamentalmente en la Alternativa Bolivariana para las Américas (ALBA).

El Software Libre está llamado a incidir notablemente en los nuevos paradigmas de la educación superior, tales como la universalización, inclusión social, formación de valores, innovación, formación masiva del capital humano, internacionalización e integración regional universitaria, virtualización y semipresencialidad de los procesos pedagógicos, entre otros. Las proyecciones del Software Libre sobre los procesos pedagógicos contemporáneos están condicionadas por los cambios ocurridos en la enseñanza universitaria en los últimos tiempos, tales como: masificación, reducción de financiamientos, formación integral, nuevos escenarios tecnológicos y pérdida de exclusividad.

Las plataformas informáticas libres brindan las mejores posibilidades para favorecer el alcance de la misión fundamental de la Educación Superior, en la preservación, desarrollo y promoción de la cultura de la humanidad, a través de los procesos sustantivos en estrecha vinculación con la sociedad. La calidad de la educación, expresada en la excelencia académica, la pertinencia integral y el pleno acceso logra una vía de realización con la utilización y desarrollo del Software Libre, precisamente porque este movimiento tecnológico propicia la formación de recursos humanos de mayor calidad y el incremento del soporte material para los procesos educativos.

Si el uso de software privativo en los procesos educacionales en algunos países capitalistas se realiza de forma gratuita, aparentando una supuesta ayuda financiera, cuando en realidad lo que hace es garantizar la adicción a este tipo de plataforma informática, para con ello proyectar seguros clientes en mercados futuros; el Software Libre ofrece una ética totalmente distinta y la posibilidad de desarrollar los procesos universitarios con seguridad y sin compromisos comerciales. La correcta aplicación del modelo semipresencial en el contexto de la universalización del conocimiento constituye uno de los mayores retos de la educación superior contemporánea y a la vez inconcebible sin la activa participación de los entornos virtuales con soporte informático en plataformas libres.

“Educación para todos durante toda la vida”, en tanto premisa fundamental de los procesos pedagógicos actuales, presupone un importante papel para la enseñanza posgraduada, donde el uso de Software Libre está llamado a jugar un rol protagónico, por sus ventajas económicas y didácticas. Cuando profundizamos en el estudio de las proyecciones sociales del Software Libre en general y sobre los procesos pedagógicos en particular, nos percatamos que esta nueva forma de organización de la ciencia y la tecnología en estrecha relación con el desarrollo social, está llamada a jugar un importante papel en las funciones fundamentales de la educación superior contemporánea.



Bibliografía utilizada

- Galli, Ricardo. Ética del software libre. Universidad de las Islas Baleares. 2005.
- Más, Jordi. Marco jurídico y oportunidades de negocios en el software libre. Revista UOC papers. No. 1, 2005.
- Pérez, Javier. Filosofía y estructura de DEBIAN. Maracay, 2006.
- Stallman R. M. Software libre para una sociedad libre. Versión 1.0. 2004.
- Colectivo de Autores. (2006) La Nueva Universidad Cubana y su contribución a la Universalización del Conocimiento. Ministerio de Educación Superior-UNESCO. Editorial Felix Varela. La Habana.
- Organización Mundial de la Propiedad Intelectual (OMPI). <http://www.cinu.org.mx/onu/estructura/organismos/ompi.htm>



Servicios telemáticos sobre GNU/Linux

Marisniulkis Lescaille Cos
mlescaille@estudiantes.uci.cu
Universidad de las Ciencias Informáticas

Introducción

Comunicarse hoy con una red o acceder archivos ubicados en otra PC, dialogar en tiempo real con personas que pueden estar muy distantes, acceder a un sitio Web, son tareas que realizamos casi a diario. Estas acciones son denominadas en términos informáticos *servicios telemáticos*.

¿Qué son los servicios telemáticos?

Telemática o teleinformática es la ciencia que trata de la conectividad a distancia entre procesos. Así que servicios telemáticos son aquellos que tienen que ver con las temáticas de acceso y procesamiento de información entre computadoras. Esto se realiza a través de redes de computadoras. Una red de computadora (también llamada red de ordenadores o red informática) es un conjunto de computadoras y/o dispositivos conectados por enlaces, a través de medios físicos (medios guiados) o inalámbricos (medios no guiados) y que comparten información (archivos), recursos (CDROM, impresoras, etc.) y servicios (e-mail, chat, juegos), entre otros. Los servicios telemáticos están orientados además, a cómo diseñar, operar y gestionar redes de manera eficiente.

A continuación se relaciona algunos de las posibilidades de los últimos utilizando software libre. El uso actual de estos servicios es inmenso. La mayoría de las empresas e instituciones en el mundo de hoy necesitan una infraestructura informática para optimizar la producción, y esta debe ser además estable y segura. La búsqueda de soluciones mediante Linux ofrece estas posibilidades. Linux está especialmente orientado al trabajo en redes pues existen numerosas aplicaciones, nativas o portadas para él, que implementan casi todos los protocolos utilizados para la comunicación, tanto el lado cliente como el servidor. Ejemplo de ello son: FTP, HTTP, TCP/IP, PPP, UUCP, SMTP, SNMP, gopher, wais, news, IPX, SMB/CIFS, POP, IMAP, etc. Gracias a esto una máquina Linux puede proveer servicios de correo electrónico, resolución de nombres, news, Web, acceso remoto, compartir recursos a través de la red, mensajería instantánea, etc.

Entre las diferentes herramientas que se pueden encontrar para disponer de estos servicios en Linux se encuentran:

Instalación y configuración de servidores:

- Web (Apache)
- Proxy (Squid)
- Correo (SendMail)
- Ftp(vsftp)
- DNS (Bind)
- Base de datos (MySQL, Postgre, Oracle y otros)
- Solución LAMP (Linux, Apache, MySQL, PHP)
- Servidor de directorio(LDAP)
- Servidor de mensajería Instantánea (Jabberd)

Servidor de Base de Datos

Los Servidores de Bases de datos (**MySQL, Postgre, Oracle**, etc.) permiten aprovechar la estabilidad y seguridad que el sistema operativo Linux le ofrece para maximizar entre otros:

- Manejo de bases de datos, ya sea desde el mismo servidor o desde aplicaciones remotas.
- Sincronización de bases de datos o la de clientes entre varios servidores.



- Configuración de varios motores de bases de datos de acuerdo con las necesidades particulares, ya sea para manejo interno o remoto.

Otros Servicios:

- **Firewalls y Filtros:** Un servidor Linux con un servicio de firewall ayuda a proteger su equipo mediante el control del tráfico de la red, ya sea en grandes redes grandes o en otras de tamaño más pequeño. Los filtros ayudan a controlar o limitar el acceso de los usuarios a determinados sitios de la red.
- **Cache:** Un servidor Proxy mejora el rendimiento de los navegadores de red, reduce los costos del acceso a la red WAN, e incrementa el control sobre el acceso a la Web al almacenar en la red local el contenido Web consultado con mayor frecuencia.

Linux como controlador primario de Dominio (PDC) a través de Samba sustituye totalmente a un servidor Windows. Es por ello que es válido afirmar que las condiciones están creadas para migrar los servicios telemáticos de cualquier sistema totalmente a Linux, y obtener los mismos beneficios que contamos hoy día con el gigante de Microsoft con una condición adicional, la exención del pago y el uso y posibilidad de modificación y adaptación a las necesidades propias de los mismos.

Conclusiones

Lo relacionado aquí demuestra que podemos contar con soluciones a la medida para nuestra configuración de redes utilizando software libre. En posteriores ediciones trataremos con más profundidad algunos de estos aspectos, teniendo en cuenta que la Universidad de las Ciencias Informáticas (UCI) se encuentra en un proceso de migración real y activo, donde toma importancia vital, para la aceptación por parte de todos de este proceso como propio, el conocimiento del intercambio entre computadoras y todos los servicios que deriva, pues este es uno de los servicios que más se utiliza en la universidad, y que más intercambio de información genera.



Servicio de streaming: video en demanda usando VLC

Alexander Martínez Fajardo
afajardo@estudiantes.uci.cu
Proyecto UNICORNIOS

VLC es un software utilizado para visualizar videos y reproducir sonidos también. Es el reproductor de media del proyecto VideoLAN VLC. Reproduce MPEG, MPEG2, MPEG4, DivX, MOV, WMV, QuickTime, MP3, Ogg / Vorbis archivos, DVDs, VCDs, y flujos multimedia de diversas fuentes de la red.

VLC también puede ser usado como un servidor de streaming ya que duplica el flujo de datos que lee y transmite a través de la red a otros clientes, o lo sirve a través de HTTP. Tiene soporte para la transcodificación al vuelo de formatos de audio y de video, ya sea para los fines de la radiodifusión o para transformaciones de formatos de películas. Soporta la mayoría de los métodos de salida de audio y video, pero se pueden añadir características mediante la instalación de los plugins de audio adicionales (vlc-plugins-esd, vlcplugins-sdl, vlc-plugins-arts) o de plugins de video (vlc-plugins-sdl, vlc-plugins-ggi, vlc-plugins-glide, vlc-plugins-svga). También hay un plugin de navegador Web (mozilla-plugin-vlc).

Primeros pasos

La forma más fácil de hacer streaming con VLC es mediante su interfaz http, la cual nos permite desde la Web publicar los videos en demanda (servicio que permite que un usuario haga una solicitud al servidor y este lo comience a reproducir pero sin descargarlo a la PC cliente). Para eso debe irse al menú del vlc "Opciones" y marcamos "Preferencias". Lo primero es marcar el cuadro de marcado que está en la parte de abajo-derecha en la ventana de Preferencias que dice "Opciones Avanzadas", es para que salgan las opciones que a un usuario normal no salen por defecto. Luego buscamos la pestaña "Interfaces principales", y dentro de ella marcamos HTTP. Nos saldrán unos textos que completar al lado, solo es necesario el de la "Dirección de Host". Aquí pondremos el ip y el puerto que utilizará la interfaz http de vlc para configurar más tarde los contenidos que vamos a visualizar.

Ejemplo:

10.33.14.106:3389

Cómo hacer el streaming

Entrando en materia, para hacer streaming, luego de haber hecho lo anterior vamos a la pestaña "Volcado de Salida" y marcamos "VOD" y le decimos a vlc cual va a ser el ip y puerto por donde saldrá el flujo de datos.

Ejemplo:

10.33.14.106:8080

Guardamos las preferencias en "Salvar" y cerramos. Abrimos el vlc nuevamente desde una terminal de consola de la siguiente forma:

```
vlc -I http
```

Se activará la interfaz http. Luego acceder a la dirección <http://10.33.14.106:3389/vlm.html> y saldrá la interfaz de control Web del vlc como servidor streaming. Después se está en condiciones de publicar las videos en demanda, llenando los campos que pide la interfaz. Para acceder al video publicado por usted es mediante la dirección rtsp:

```
rtsp://10.33.14.106:8080/nombre_del_recurso
```

Nota: Debe cambiarse el ip y puerto de las diferentes configuraciones en dependencia de sus necesidades reales.



Solución Integrada de la implantación de un sistema de perfiles centralizados

Amaury Viera Hernández
avhernandez@estudiantes.uci.cu
Proyecto UNICORNIOS

Introducción

En el mundo informático de hoy las compañías, empresas, consorcios, instituciones, etc, cada vez más están dirigidas a la satisfacción de las comodidades y necesidades de cada profesional, para lograr una excelente armonía entre trabajo a realizar/satisfacción del profesional, conllevando a lograr mejores resultados productivos y elevando la calidad de lo producido.

Este pequeño pero concreto artículo aborda este tema desde la necesidad que tenía el proyecto productivo UNICORNIOS de implantar un sistema de perfil centralizado para así resolver varias problemáticas que se daban en el laboratorio, así como otras un poco más críticas.

Este artículo no pretende ser una guía exacta para realizar dicha implantación, sino un apoyo dado las necesidades de cada cual que lo vaya a realizar. El objetivo de esta documentación es lograr hacer que los usuarios compartan un home remoto por NFS. Para ello los usuarios serán almacenados en un servidor OpenLDAP. A continuación se expone lo necesario para lograr esta meta.

Servidor

Distribución: Debian Sarge o Debian Etch

Servidores a instalar:

DNS:

Herramienta: BIND9

OpenLDAP:

Herramienta: Slapd

Servidor de NFS:

Herramientas: nfs-kernel-server, nfs-common, portmap, libnss-ldap, libpam-ldap

Servicio de Control de Cuotas:

Herramienta: Quota

Cliente

Librerías para autenticar al cliente con OpenLDAP:

Herramientas: libnss-ldap, libpam-ldap, nfs-common, portmap

Instalación y configuración de los servicios necesarios

Los ficheros y archivos de configuración que aquí se mencionan, así como dos video tutoriales que explican cómo configurar las librerías pam para conectarse a un servidor ldap y como crear una configuración inicial para configurar un servidor LDAP, se encuentran en el servidor de documentación de la UCI (sólo accesible desde la UCI)³¹

DNS:

En cualquier documentación podrá encontrarse cómo instalar un servidor de dominio, para el cual se instalará bind9 y dnsutils.

```
apt-get install bind9 dnsutils
```

³¹ http://10.128.50.121/Software/Instaladores/Home_Compartido/



Sin remitirse a alguna documentación, sólo con instalar estos dos paquetes y copiar los archivos que están dentro `dns.tar.gz` hacia la carpeta `/etc/bind`, los archivos `db.127.facx.uci.cu` y `db.facx.uci.cu`, tendrán el nombre del DNS que se vaya a utilizar, y dentro se editarán según la dirección IP que tendrá el servidor de nombre. En este caso el IP del DNS es **10.33.20.214**. Solo habría que editar estos archivos y dentro de ellos especificarle además el nombre de la nueva zona de dominios.

OpenLDAP:

Para ello se instala el paquete **slapd**

```
apt-get install slapd
```

Durante la instalación, al pedir el DNS a usar se utilizará el previamente creado, tipo de base de datos a usar `bdb` y demás parámetros por defecto. En ocasiones, al instalarlo necesita un paquete llamado **db4.2-utils** o **db4.3-utils**, entonces se escribe el comando:

```
/etc/bash_completion
```

```
apt-get install db4
```

Luego se presiona `tab` dos veces y aparecerán todos los paquetes cuyo nombre comience así, y se selecciona el deseado (es decir el tipo de base de datos que necesite). Se sugiere instalar una herramienta para administrar el servidor LDAP, la mejor de ellas es el `phpldapadmin`. Para ello inicialmente instalar el `apache`, y luego directamente el `phpldapadmin`, donde al solicitar qué servidor Web usará, se pone `apache`.

Una vez instalado correctamente el `phpldapadmin`, se accederá a la interfaz de administración del `Ldap`, a través de la url **`http://ip/phpldapadmin`**, puede darse cuenta que solicitará un `password`, el cual no es más, que el `password` que previamente se dijo era el del servidor `Ldap`. Una vez dentro, se debe crear un objeto de tipo `ou` (Organizational Unit), es decir dar en `create new entry here`, y seleccionar `Organizational Unit`, se presiona en `proceed`, lo que llevará a ponerle el nombre a la unidad organizacional, se escribe el nombre, y ya estará creada una unidad organizacional, en la cual se almacenarán todos los usuarios que tendrá el servidor `Ldap`, es decir, sería repetir el paso anterior, pero adicionando objetos de tipo `User Account` (`posixAccount`), la otra `ou` `Posix Group` y dentro de ella se crean grupos de forma similar a lo anterior con el nombre de cada grupo, y a cada uno se le adiciona cuantos atributos `Member uid` tenga en el servidor `Ldap`.

También puede añadirse un primer fichero `ldif`, indicándole que se crearán dos unidades organizacionales. Para ellos consultar el fichero llamado `ou.ldif`, el cual al importarlo desde el `phpldapadmin` creará dos unidades organizacionales, una de ellas para añadir personas, es decir, objetos de tipo `User Account` (`posixAccount`), y ahí se introducen todas las características del usuario que creado, entre ellas un `uid` (único para cada usuario), un `gid` (común para cada usuario), y un `home directory` (en este especificar dónde estará el `home` de cada usuario).

En la otra unidad organizacional añadir los grupos que se necesitan para que los clientes al conectarse al `Ldap`, puedan escuchar música, reproducir videos, montar floppy, cdrom y flash. Lo primero es crear los grupos, que son objetos de tipo `Posix Group`, diciéndole el nombre del grupo, ejemplo para el sonido se pondrá `audio` (este nombre se escoge arbitrariamente), ese grupo tiene un `gid`, el cual debe ser el `gid` que use la máquina cliente para el grupo de `audio`, para ello se debe instalar una serie de scripts en `perl`, los cuales se pueden lograr instalando el programa que se encuentra en el repositorio llamado `migrationtools` (esto debe ser en la máquina cliente), e ir a `/etc/group` donde guarda todos los grupos que usa el sistema y cuál es su `gid` (un número), este número es el que se pondrá a medida que vaya creando grupos de `audio` en el servidor `Ldap`, en la unidad organizacional grupos, luego se añade un objeto de tipo `memberUid` (al grupo creado) y añadir a él los usuarios que se necesitan pertenezcan a ese grupo.

Servidor de NFS:

Lo primero en este paso será instalar el paquete **`nfs-kernel-server`**, y los demás que se mencionan en la sección de herramientas necesarias. Para configurar el servidor como `home` compartido sólo hay que ir al fichero `/etc/exports` donde se indica al servidor de `nfs` que exporte en este caso para la subred `10.33.20.0`, el fichero quedaría así:

```
/home 10.33.20.0/255.255.255.0(rw)
```



Definir permisos en el servidor

Se debe crear manualmente una carpeta, (lo cual puede ser perfectamente automatizable) con el nombre de cada usuario (debe coincidir el nombre de usuario con el de la carpeta) que se tiene almacenado en el servidor ldap, la misma debe tener los permisos, 755, los cuales se asignarán:

```
chmod 755 [nombre]
```

Además indicar que ese usuario sólo será dueño de esa carpeta que será la que funcione como su home, por razones de seguridad, esto se logra de la siguiente manera:

```
chown -R usuario:grupo usuario/
```

Usuario es el usuario creado en el ldap y que es el nombre de la carpeta a la que se le está definiéndole el dueño, y **grupo**, es el grupo que anteriormente mencionado que sería común para todos los usuarios.

Imagen cliente

Para que la imagen cliente funcione correctamente debe tener instalado los paquetes mencionados con anterioridad. La configuración de las librerías libnssldap libpam-ldap, debe ser así:

1. LDAP Server host	10.33.20.247
2. Distinguished name of the search base	dc=facx,dc=uci,dc=cu
3. LDAP version to use	3
4. Make local root Database admin	No
5. Database requires logging in	No
6. Local crypt to use when...	md5

Nota: para poder configurarlas bien se debe reconfigurar pues no aparece la opción Local crypt... para ello sería teclear el siguiente comando:

```
dpkg-reconfigure libnss-ldap libpam-ldap
```

Una vez hecho esto se estaría indicando a la imagen cliente qué servidor ldap(OpenLdap) usar cuando vaya a autenticarse. Luego sería en el fstab de la máquina cliente montar el home remoto, para que este se monte automáticamente cada vez que inicie el sistema, se debe añadir lo siguiente en el fichero /etc/fstab:

```
10.33.20.247:/home /home/ nfs rw,defaults 00
```

Es aconsejable, añadir un script para que las particiones se recarguen automáticamente al inicio del sistema, esto sería un pequeño script en bash que ejecute el comando mount -a cada vez que el sistema arranque, el script se encuentra en el archivo script.tar.gz, el cual debe copiarse hacia /etc/rcS.d con el nombre SXscript, donde x es un número que no se encuentre ya entre los demás enlaces simbólicos que hay en ese directorio y además tener permisos de ejecución.

Nota: Estas dos librerías es necesario instalarlas y configurarlas en el servidor para que este también autentique contra el OpenLdap.

Para comprobar que todo ha funcionado correctamente debe teclearse la id usuario y esto debe decir quién es este usuario dentro del ldap es decir cual es su id, y a qué grupos pertenece, si sucede es que se encuentran funcionando correctamente el servidor ldap y que las librerías del cliente para autenticarse contra el servidor ldap.

Servicio de cuotas

Se hace necesario también establecer un sistema de cuotas para los usuarios debido al alto número de usuarios y el poco espacio disponible en el servidor. Para ello instalar cuotas. Para establecer la partición que se controlará mediante cuotas dirigirse al directorio raíz de la partición y ejecutar:

```
$ touch aquota.user  
$ touch aquota.group
```



Luego modificar en el `/etc/fstab` la partición a la cual se le establecerá la cuota para garantizar que el sistema realice periódicamente comprobaciones sobre ella.

```
...  
/dev/sda5 /home ext3  
defaults,usrquota,grpquota 0 2  
...
```

Al recargar la partición esta estará ya lista para comprobar cuotas. Para editar la cuota de un usuario:

```
$ edquotas <usuario>  
Disk quotas for user teratux (uid 5112):  
Filesystem blocks soft hard inodes /dev/sda5 685348 0 800000 28473 soft hard 0 0
```

#Blocks nos dirá la cuota consumida y Hard el máximo que puede tener. Para hacer extensivo el sistema de cuotas de un usuario al resto emitir estos dos comandos:

```
$ getent passwd|grep home|cut -d":" -f1 > logins.txt  
$ for x in `cat logins.txt`;do edquota -p <usuario_base> $x;done
```

Iniciar el servicio de cuotas:

```
$ quotaon -a
```

Una vez configurado el servicio de cuotas debe garantizarse que sea accesible desde las carpetas remotas. Aquí entra a jugar `remote-quotas` o `rquota`. Simplemente se edita (una vez instalado `rquota`) el fichero `/etc/rpc` y escribir la línea:

```
$ rquotad 100011 rquotaprogram quota rquota
```

Reiniciar el servicio de `rquota` y listo. En ocasiones se hace necesario ejecutar el comando **`quotaoff -a`** y luego el comando **`quotaon -a`** y luego reiniciar el servicio **`/etc/init.d/quota restart`** y **`/etc/init.d/quotarpc restart`**.

Paso Final

Una vez logrado lo anterior se quiere lograr que los clientes al entrar por primera vez al sistema tengan una configuración inicial del entorno de escritorio previamente seleccionado. Entonces se toma la configuración inicial de un usuario y hacerle un compactado. Para ello, si el usuario se llama pepe por ejemplo, debe posicionarse dentro de su home y listar el comando:

```
cd /home/pepe  
tar-cvzf/[dirección]/skel.tar.gz *3
```

De esa manera se tendrá en **[dirección]** el compactado con la configuración que se desea tengan todos los usuarios. Esto se hace muy necesario porque podría añadirse en el navegador las bookmarks necesarias a enlaces de sitios del proyecto, como track, intranet, servidores Web y demás, además de varios softwares que necesitan ser referenciados a través de lanzadores para poder ejecutarlos y con esto se evita la enorme molestia al usuario de memorizar dónde está instalado el software y tener que ejecutarlo para poder usarlo.

Después de haber creado el compactado, correspondería crear la carpeta con los permisos necesarios y para ello se ejecutará el script que se encuentra dentro del compactado `create_home.tar.gz`, este archivo al ejecutarlo debe editarse, con cualquier editor de texto, en este caso donde está el 10000 es debido a que en el `chown` se le dice que el usuario a crear tendrá el `gid` 1000 y es muy posible que se quiera tener otro `gid`, y además editar el parámetro `/home` en el caso que no se quieran almacenar los usuarios en `/home`, lo cual no es recomendable, por cuestiones de organización.

Luego sería solo ejecutar el script pasándole como parámetros el nombre del usuario a crear y el skel o compactado del cual heredará su configuración. Para ello posicionarse por la consola con `cd` en el directorio



donde se encuentra el script y que además para mayor comodidad debe estar el compactado y creamos el home de cuántos usuarios se desean a través de:

```
./create_home usuario skel.tar.gz
```

Con esto se evita estar definiendo permisos usuario por usuario y dándole un chown, además que se crea una configuración inicial para ese usuario una vez que acceda por primera vez al sistema. Es necesario aclarar, que sería bueno no dejar ninguna aplicación corriendo en el momento de crear el compactado que se use como base, así como no cerrar abruptamente la sesión de este, debido que esto se guarda en el home del mismo y entonces los usuarios a los que se le cree el home pudieran surgirle incomodidades al iniciar la sesión como el firefox pidiendo ser restaurado o un procesador de texto pidiendo recuperar la información no guardada al cerrarse abruptamente.

Consejos útiles para el uso del home compartido

- Siempre cerrar la sesión adecuadamente.
- Siempre cerrar las aplicaciones adecuadamente, es decir no cerrar la sesión con aplicaciones corriendo.
- Nunca reiniciar el ordenador presionando el botón de restart.

Posibles problemas a presentarse y su solución

Mensaje que dice que el firefox se está ejecutando: Para ello borrar el archivo .parentlock que se encuentra dentro de la carpeta .mozilla dentro del home. Igualmente se puede borrar la carpeta .mozilla, pero esto implicaría que se elimine la configuración.

En general, cualquier aplicación que muestre la salida “está ejecutándose”, sencillamente se borra del home y automáticamente volverá a iniciarse en el ordenador la próxima vez que sea ejecutada.

Hasta aquí se ha instalado y configurado un servidor de home compartido por nfs que usa un servicio de quotas para los usuarios y para autenticarse los clientes usan un servidor ldap. Esta documentación ha sido proporcionada por el proyecto Scada, con la colaboración incondicional de Roberto Alejandro Espi Muñoz, sin el cual nada de lo logrado en el proyecto UNICORNIO de la facultad 10 hubiera sido posible.



GIMPmanía

David Padrón Alvarez
dpadron@estudiantes.uci.cu
Proyecto GRaph Tool,
Facultad de Bioinformática y Gestión Biomédica

El Diseño Gráfico asistido por computadoras, es actualmente un eslabón fundamental en el desarrollo de software y soluciones computacionales que requieran el trabajo con imágenes. La generalización de aplicaciones privativas como Adobe Photoshop ha provocado que se cree un patrón de uso alrededor de las mismas; sin embargo, sus elevados precios permiten que las alternativas de diseño con Software Libre (SWL) como GIMP (GNU Image Manipulation Program) se coloquen en una posición ventajosa con respecto a las de software propietario.

El origen del GIMP fue en el año 1996, en la Universidad de Berkeley, California; y es considerado por muchos como prueba del proceso de desarrollo del Software Libre, pues demostró que se pueden crear aplicaciones que los usuarios no avanzados puedan usar productivamente. El GIMP abrió el camino a otros proyectos como KDE, GNOME, Mozilla Firefox, Open Office y otras aplicaciones posteriores. La librería GTK creada para él, es junto a Qt, el toolkit de creación de interfaces gráficas más potente en el mundo del SWL. GIMP, junto a Blender, Inkscape y otras aplicaciones, ha sido capaz de convertir a GNU/Linux en una plataforma de Diseño Gráfico.

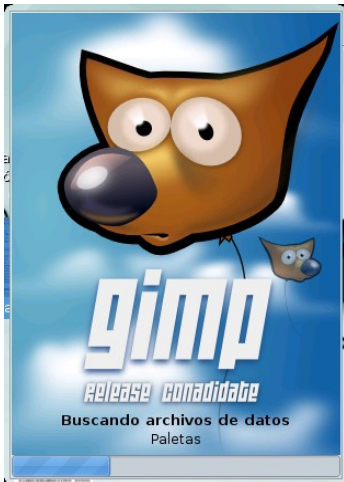


Figura 39. Splash de GIMP

Pero la ventaja ya no es sólo respecto al precio; las potencialidades del GIMP han demostrado que puede sustituir completamente al Photoshop, cubriendo en un 98% las necesidades de los diseñadores, y el 2% restante, se refiere a los plugins de terceros que han sido hechos para Photoshop y que, producto al uso masivo de esta herramienta, superan en número a los hechos para GIMP. Pero aún para este 2% GIMP tiene una alternativa: el PSPI. El PSPI es un plugin de GIMP escrito por Tor Lillqvist. Esta extensión da soporte para muchísimos de los plugins de Photoshop. El PSPI se debe copiar en la carpeta de plugins del GIMP y se muestra en el Menú Exts. Al utilizarlo por primera vez se le debe especificar una carpeta donde estarán los archivos .8bf, luego podrá ser usado cualquier plugin de terceros que se copie allí.

El GIMP ha entrado en una nueva fase en su desarrollo; la salida de su versión 2.4 estable, representa un gran avance en cuanto a usabilidad, pues han sido rediseñadas las ubicaciones de los menús, trae incluido el Algoritmo de selección SIOX (Simple Interactive Object Extraction), el clonado en perspectiva, la brocha de curado (healing brush), la posibilidad de copiar de archivos SVG, los pinceles dinámicos con posibilidad de variaciones de tamaño y otros retoques en general para el programa. GIMP se encuentra disponible para



plataformas GNU/Linux, MacOS y Windows. La última versión liberada en la rama estable, hasta el momento en que se escribe este artículo, es la 2.4.2 con alrededor de 30 bugs corregidos. Estas imágenes representan algunos resultados de lo que se puede hacer con esta poderosa herramienta:



Figura 40. Contorno 3D

Figura 41. Texto con manchas

En la UCI, el uso del GIMP constituye un paso de avance en el proceso de Migración de todo el país, pues con su uso se eliminan los problemas que suponen el uso de herramientas privativas cuando se deben comprar las Licencias. El desarrollo de los proyectos productivos usando herramientas libres bajo licencia GPL o similares, significa un ahorro considerable para la economía en general. Proyectos cinematográficos como Plumíferos, Shreck y Elephant Dreams lo han usado para sus texturas y mejoras de imágenes.

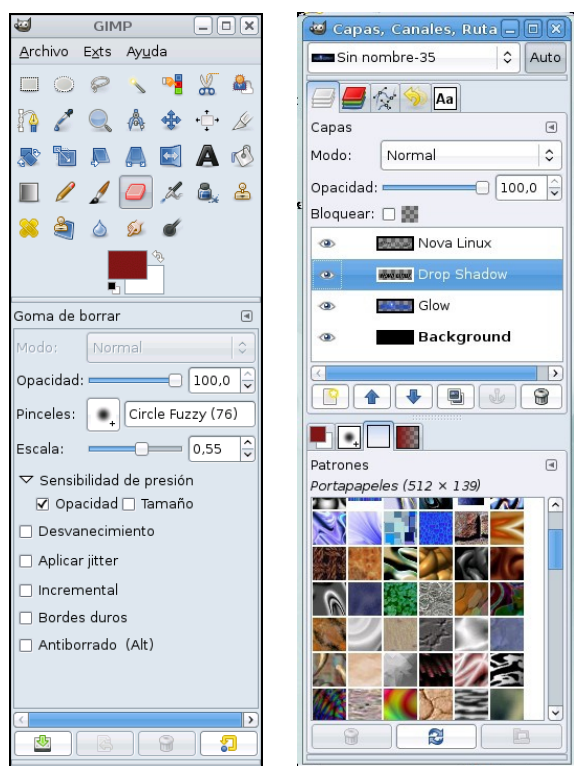


Figura 42. Ventanas de GIMP

Nuestra comunidad aún es joven, pero la cantidad de usuarios de GNU/Linux crece a diario exponencialmente; por eso debemos estar preparados para ayudar y guiar a los nuevos que van llegando. La falta de documentación no debe ser un obstáculo; si cada cual documenta lo que sabe, o adapta los tutoriales de Photoshop al entorno de GIMP, pronto podremos ser tantos gimperos como usuarios hay de Photoshop. Dar el paso es lo más importante. El píxel libre, espera por nosotros.



Creación y uso de librerías dinámicas

Dayron Pérez Roldán
droldan@estudiantes.uci.cu
Proyecto UNICORNIOS

Resumen

Las librerías son una forma sencilla y potente de modularizar y reutilizar código. En este artículo se abordará el proceso de creación y uso de librerías dinámicas. Se desarrollará una librería para realizar conversiones de un tipo de variables a otro, 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++.

Introducción

Según vamos haciendo programas de ordenador, nos damos cuenta que algunas partes del código se utilizan en muchos de ellos. Por ejemplo, podemos tener varios programas que utilizan números complejos y las funciones de suma, resta, etc son comunes. Sería estupendo poder meter esas funciones en un directorio separado de los programas concretos y tenerlas ya compiladas, de forma que podamos usarlas siempre que queramos.

Las ventajas enormes de esto son:

- No tener que volver a escribir el código (o hacer copy-paste).
- Nos ahorraremos el tiempo de compilar cada vez ese código que ya está compilado. Además, ya sabemos que mientras hacemos un programa, probamos y corregimos, hay que compilar entre muchas y "más muchas" veces.
- El código ya compilado estará probado y será fiable. No las primeras veces, pero sí cuando ya lo hayamos usado en 200 programas distintos y le hayamos ido corrigiendo los errores.

La forma de hacer esto es hacer librerías. Una librería son una o más funciones que tenemos ya compiladas y preparadas para ser utilizadas en cualquier programa que hagamos, y deberías tratar de crearlas lo mas estándar posible. En linux podemos hacer dos tipos de librerías: **estáticas** y **dinámicas**.

Una librería estática es una librería que "se copia" en nuestro programa cuando lo compilamos. Una vez que tenemos el ejecutable de nuestro programa, la librería no sirve para nada (es un decir, sirve para otros futuros proyectos). Podríamos borrarla y nuestro programa seguiría funcionando, ya que tiene copia de todo lo que necesita. Sólo se copia aquella parte de la librería que se necesite. Por ejemplo, si la librería tiene dos funciones y nuestro programa sólo llama a una, sólo se copia esa función.

Una librería dinámica No se copia en nuestro programa al compilarlo. Cuando tengamos nuestro ejecutable y lo estemos ejecutando, cada vez que el código necesite algo de la librería, irá a buscarlo a ésta. Si borramos la librería, nuestro programa lanzará un error de al no encuentra.

Desarrollo

Como ya se ha ido mencionando en este artículo abordaremos el tema de las librerías dinámicas y para esto pasaremos a desarrollar un ejemplo, que sirva para guiar el proceso de aprendizaje. El ejemplo que se desarrollará a continuación incluye trabajo con conversiones de tipos en C++.

Lo primero será crear un nuevo proyecto en el Code::Blocks para poder realizar el programa deseado: File/New/Project/Console application (Figura 43).

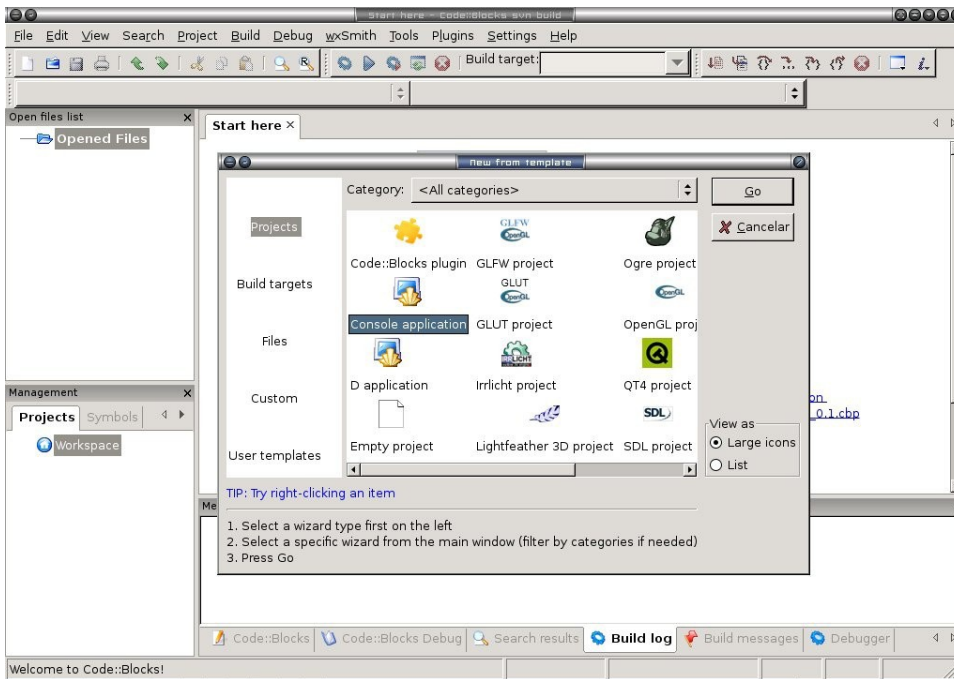


Figura 43. Creación de un nuevo proyecto

Luego pasaremos al asistente para la creación del proyecto, donde daremos nombre al mismo así como la dirección donde se guardara (Figura 44), luego tan solo presionaremos Next a todas las demás opciones.

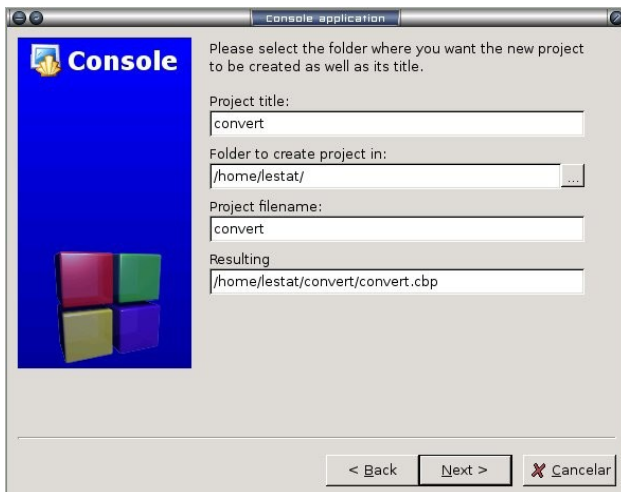


Figura 44. Asistente de creación del proyecto

Ahora bien, vamos a comenzar a programar. Lo primero que debemos hacer es crear un fichero de cabecera (.h) que se llamara “_convert.h” con aquellas funciones que se desean incluir en la librería, en nuestro caso será así :

```
#ifndef _CONVERT_H_INCLUDED
#define _CONVERT_H_INCLUDED
#include <iostream>
/*Convertir de Cadena a Entero*/
```



```
void StrToInt(char*, int&);  
/*Convertir de Cadena a Float*/  
void StrToFloat(char*, float&);  
  
/*Convertir de Entero a Cadena*/  
void IntToStr(int, char*&);  
/*Convertir de Float a Cadena*/  
void FloatToStr(float, char*&);  
#endif // _CONVERT_H_INCLUDED
```

Un detalle importante a tener en cuenta, son los **#define** del fichero de cabecera (.h). Al hacer una librería, no sabemos en qué futuros programas la vamos a utilizar ni cómo estarán organizados. Supongamos en un futuro programa que hay un fichero de cabecera **fichero1.h** que hace **#include** del nuestro. Imaginemos que hay también un **fichero2.h** que también hace **#include** del nuestro. Finalmente, con un pequeño esfuerzo más, imaginemos que hay un tercer **fichero3.c** que hace **#include** de **fichero1.h** y **fichero2.h**.

Cuando compilemos **fichero3.c**, dependiendo de lo que haya definido en **libreria1.h**, obtendremos un error. El problema es que al incluir **fichero1.h**, se define todo lo que haya en ese fichero, incluido lo de **libreria1.h**. Cuando se incluye **fichero2.h**, se vuelve a intentar definir lo contenido en **libreria1.h**, y se obtiene un error de que esas definiciones están definidas dos veces.

La forma de evitar este problema, es meter todas las definiciones dentro de un bloque **#ifndef - #endif**, con el nombre (**_LIBRERIA_1_H** en el ejemplo) que más nos guste y distinto para cada uno de nuestros ficheros de cabecera. Es habitual poner este nombre precedido de **_**, acabado en **_H** y que coincida con el nombre del fichero de cabecera, pero en mayúsculas.

Dentro del bloque **#ifndef - #endif**, hacemos un **#define** de ese nombre (no hace falta darle ningún valor, basta con que esté definido) y luego definimos todos nuestros tipos y prototipos de funciones. Cuando incluyamos este fichero por primera vez, **_LIBRERIA_1_H** no estará definido, así que se entrará dentro del bloque **#ifndef - #endif** y se definirán todos los tipos y prototipos de funciones, incluido el mismo **_LIBRERIA_1_H**. Cuando lo incluyamos por segunda vez, **_LIBRERIA_1_H** ya estará definido (de la inclusión anterior), por lo que no se entrará en el bloque **#ifndef - #endif**, y no se redefinirá nada por segunda vez.

Es buena costumbre hacer esto con todos nuestros .h, independientemente de que sean o no para librerías. Si te fijas en algún .h del sistema verás que tienes este tipo de cosas hasta aburrir. Después de aclarado algunos puntos importantes, pasaremos a implementar las funciones de la librería, para ello se deberá adicionar un nuevo fichero al proyecto que se llamara “_convert.cpp”.

```
#include "_convert.h"  
void StrToInt(char* c, int& i)  
{  
    i = atoi(c);  
}  
void StrToFloat(char* c, float& f)  
{  
    f = atof(c);  
}  
void IntToStr(int i, char*& c)  
{  
    sprintf(c, "%2d", i);  
}  
void FloatToStr(float f, char*& c)  
{  
    sprintf(c, "%6.2f", f);  
}
```



}

Una vez que tengamos implementadas las funciones, pasaremos a compilar el proyecto, acción mediante la cual se obtendrán los ficheros objetos del proyecto (.o). En este caso en particular obtendremos el código objeto del “main.cpp” y del “_convert.cpp” pero el que nos importa a nosotros es el código objeto generado como resultado de compilar el código del “_convert.cpp” --> “_convert.o”.

Creación de la librería dinámica

La creación de librerías dinámicas corre a cargo del enlazador o montador (en nuestro caso el ld) aunque también es posible indicar al compilador las opciones necesarias para el montaje y de ese modo, será él quien se encargue de comunicárselas al montador. Cuando se crea un objeto dinámico es necesario que dicho código objeto sea independiente de la posición, para conseguir este tipo de código debe especificarse al compilador la opción -fPIC (Position Independent Code). Dicho flag debe indicarse tanto en la compilación (Figura 45) como en el montaje de la librería.

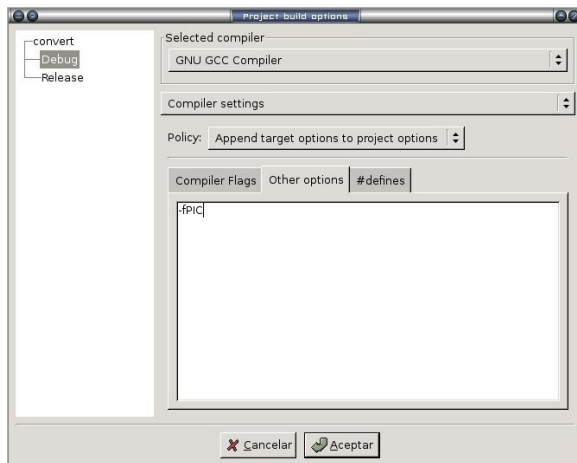


Figura 45. Especificar flag de compilación

O podemos hacerlo compilando el proyecto por la consola mediante los comandos siguientes:

```
$ gcc -fPIC -c -o _convert.o _convert.cpp
```

Para montar los objetos es necesario además indicar la opción -shared para que el resultado sea un fichero objeto “compartible”.

```
$ gcc -shared -fPIC -o libconvert.so _convert.o
```

En este caso, la librería tiene como extensión .so que significa shared object. Para utilizar esta librería desde un programa debemos colocar en /usr/include/ el archivo de cabecera que creamos en el proyecto “_convert.h”, y colocar también la librería dinámica “libconvert.so” en /usr/lib/ luego a crear un nuevo proyecto y probar la librería. Un ejemplo sencillo quedaría así:

```
#include <iostream>
#include <_convert.h>
using namespace std;
int main()
{
    int i;
    StrToInt("34", i);
    cout<< ++i;
```



```
return 0;
```

```
}
```

Ah, casi se olvida; no basta solo con colocar el fichero de cabecera en `/usr/include` sino también que se debe incluir la librería dinámica en el proyecto nuevo. En Code::Blocks bastaría con ir a `Project/BuildOptions/LinkerSettings/Add` y se buscara “`libconvert.so`” en `/usr/lib/`. Como se indica en la Figura 46.

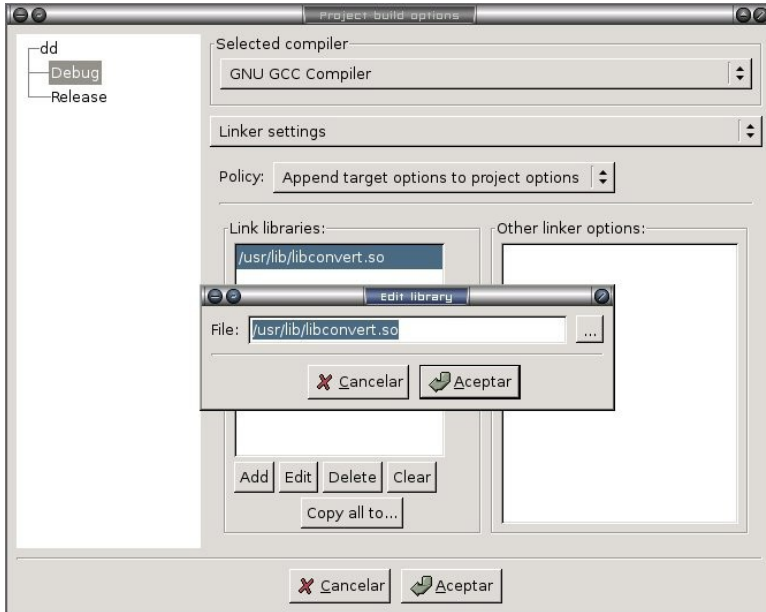


Figura 46. Adicionar librería dinámica

Usando la librería dinámica

Al hacer uso de una librería, el compilador busca primero una versión dinámica (.so), si no la encuentra entonces busca una estática. Si se tienen las dos versiones de una librería y se quiere utilizar la versión estática debe indicarse al montador el flag `-static`.

Cuando un programa utiliza librerías dinámicas necesita localizarlas en tiempo de ejecución (al contrario que con las librerías estáticas). Los lugares donde un programa busca las librerías dinámicas son los siguientes (en este orden):

1. En los directorios de la variable `LD_LIBRARY_PATH`.
2. En el fichero `ld.so.cache`
3. En los directorios `/usr/lib` y `/lib`
4. En los directorios contenidos en `ld.so.conf`

Si el programa no encuentra la librería que necesita imprimirá un mensaje de error con el siguiente aspecto:

```
$ ./ejem_convert
ejem_convert: error in loading shared libraries:
libfich.so: cannot open shared object file: No such file or directory
```

Normalmente, lo más adecuado, es utilizar la variable de entorno `LD_LIBRARY_PATH` para indicar en qué directorios debe buscar: `export LD_LIBRARY_PATH = $LD_LIBRARY_PATH : /home/lestat/convert/`



Conclusiones

De manera general se han abordado algunos rasgos principales del tema de las librerías dinámicas, las cuales resultan de suma importancia para aquellos que desean hacer sus aplicaciones lo mas pequeñas posibles. Espero todos puedan comenzar a crear sus propias librerías y recuerden Documentarlas lo mejor posible para que sean de uso popular por la Comunidad.



Programación de socket en Linux con C++

Dayron Pérez Roldan
droidan@estudiantes.uci.cu
Proyecto UNICORNIOS

Resumen

Los sockets representan una interfaz de comunicación que ofrece un mecanismo de interacción general entre dos procesos cualesquiera, que pertenezcan a un mismo sistema o a dos diferentes. En este artículo se abordarán los aspectos más importantes de la programación de sockets, utilizando Code::Blocks como entorno de programación para escribir el código, el cual podría ser escrito en cualquier IDE (Entorno de Desarrollo Integrado) que soporte C++. Este artículo no pretende enseñar a programar socket, tan sólo es una panorámica de esa programación, ilustrada con pequeños ejemplos.

Introducción

La programación de sockets muchas veces tiende a ser algo complicado, pero cuando llegamos a entenderla puede ser realmente de mucha utilidad. Un socket, explicado de manera simple, se puede decir que es una manera de comunicarse con otra computadora o con la nuestra. Para ser más exacto, es una manera de comunicarse usando descriptores de ficheros estándar de Unix. En Unix, todas las acciones de entrada y salida son desempeñadas escribiendo o leyendo en uno de estos descriptores de fichero, los cuales son simplemente un número entero, asociado a un fichero abierto que puede ser una conexión de red, un terminal, o cualquier otra cosa (recuerde uno de los primeros conceptos que aprendemos al usar UNIX "En un sistema UNIX, todo es un fichero").

Muchas veces se programan aplicaciones de escritorio y gustaría que las mismas pudieran comunicarse entre sí, desde diferentes puntos en toda una intranet o incluso internet; como por ejemplo un pequeño chat para conversar con los amigos, o quizás se desarrolla un juego y quiere aportarle la habilidad de ser jugado entre dos personas desde diferentes extremos de la red, etc.

Pueden surgir muchos usos para los sockets y todo se va a resumir en que en ellos reside la herramienta básica orientada al paso de mensajes entre procesos para programar en red sobre TCP/IP. Ahora bien, sobre los diferentes tipos de sockets se puede hacer mención de tres de ellos:

- SOCK_STREAM (Sockets de Flujo)
- SOCK_DGRAM (Sockets de Datagramas)
- SOCK_RAW (Sockets Puro, solo usados por el root)

No quiere decir que estos sean los únicos sockets que existen; no obstante en este artículo sólo se hablará de los sockets de flujo y los sockets de datagramas.

Sockets de flujo

Generalmente no poseen errores. Por ejemplo si se enviara por el socket tres objetos "A, B, C" llegarán a su destino en el mismo orden. Estos sockets usan TCP (Transmission Control Protocol) y es este protocolo el que asegura el orden de los objetos durante la transmisión. Un ejemplo de aplicación que usa socket de flujo es **telnet**.

Sockets de datagrama

Estos usan UDP ("User Datagram Protocol"), y no necesitan de una conexión accesible como los Sockets de Flujo, tan sólo se construirá un paquete de datos con información sobre su destino y se enviará afuera, sin necesidad de una conexión. Un ejemplo de aplicaciones que usan socket de datagramas son **tftp** y **bootp**. Mucho más podría explicarse aquí sobre estas dos clases de sockets, pero esto es suficiente como para captar el concepto básico de socket. Entender qué es un socket y algo sobre estos dos tipos de sockets de Internet es un buen comienzo, pero lo más importante será saber cómo trabajar con ellos.



En la programación con sockets existen estructuras de datos que son las encargadas de almacenar la información del socket así como la descripción del nodo remoto entre otras funcionalidades. Una estructura de dato es un tipo de dato que puede contener otros tipos de datos y los cuales son agrupados en un único tipo definido por el usuario.

Veamos entonces las siguientes estructuras:
Contiene información del Socket.

```
struct sockaddr
{
    unsigned short sa_family;
    char sa_data[14];
};
```

Hace referencia a los elementos del Socket.

```
struct sockaddr_in
{
    short int sin_family;
    unsigned short int sin_port;
    struct in_addr sin_addr;
    unsigned char sin_zero[8];
};
```

```
struct in_addr
{
    unsigned long s_addr;
};
```

Almacena información del nodo remoto. Esto es en el cliente.

```
struct hostent
{
    char *h_name;
    char **h_aliases;
    int h_addrtype;
    int h_length;
    char **h_addr_list;
    #define h_addr h_addr_list[0]
};
```

Está definida en **#include <netdb.h>**. Las demás librerías que se deben incluir para programar socket son:

```
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
```

Conversiones

Existen dos tipos de ordenamiento de bytes: bytes más significativos, y bytes menos significativos. Este es llamado "Ordenación de Bytes para Redes", algunas máquinas utilizan este tipo de ordenación para guardar sus datos internamente. Existen dos tipos a los cuales seremos capaces de convertir: short y long.

htons() : "Nodo a variable corta de Red"
htonl() : "Nodo a variable larga de Red"
ntohs() : "Red a variable corta de Nodo"



ntohl() : "Red a variable larga de Nodo"

Direcciones IP

Existen algunas funciones que ayudarán a manipular direcciones IP, como son **inet_addr()** y **inet_ntoa()**. Por un lado, la función **inet_addr()** convierte una dirección IP en un entero largo sin signo (**unsigned long int**), por ejemplo:

```
dest.sin_addr.s_addr=inet_addr("10.33.3.6")
```

Por otro lado, **inet_ntoa()** convierte a una cadena que contiene una dirección IP en un entero largo. Por ejemplo:

```
char *ip;  
ip=inet_ntoa(dest.sin_addr);  
printf("La dirección es: %s\n",ip);
```

Nota: La función inet_addr() devuelve la dirección en formato de Ordenación de Bytes para Redes por lo que no necesitaremos llamar a htonl().

Vistos ya elementos de importancia con respecto a socket se pueden mencionar las llamadas al sistemas mas importantes.

Funciones importantes en el uso de socket

Socket: Devuelve un descriptor de socket, el cual podrá usarse luego para llamadas al sistema.

```
#include <sys/types.h>  
#include <sys/socket.h>  
int socket(int domain,int type,int protocol);
```

Bind: Su función esencial es asociar un socket con un puerto (de nuestra máquina).

```
#include <sys/types.h>  
#include <sys/socket.h>  
int bind(int fd, struct sockaddr *my_addr ,int addrlen);
```

Connect: Se usa para conectarse a un puerto definido en una dirección IP.

```
#include <sys/types.h>  
#include <sys/socket.h>  
int connect(int fd, struct sockaddr *serv_addr, int addrlen);
```

Listen: Se usa para esperar conexiones entrantes. Después de llamar a listen(), se deberá llamar a accept(), para así aceptar las conexiones entrantes.

```
#include <sys/types.h>  
#include <sys/socket.h>  
int listen(int fd,int backlog);
```

Accept: Se usa para conseguir establecer la conexión entrante.

```
#include <sys/types.h>  
#include <sys/socket.h>  
int accept(int fd, void *addr, int *addrlen);
```

La secuencia resumida de llamadas al sistema es:

```
socket()  
bind()  
listen()  
accept()
```

Ahora se hablará de las funciones que se usan para comunicarse a través de socket.



Send: Se usa para enviar datos.

```
#include <sys/types.h>
#include <sys/socket.h>
int send(int fd, const void *msg, int len, int flags);
```

Recv: Se usa para recibir datos.

```
#include <sys/types.h>
#include <sys/socket.h>
int recv(int fd, void *buf, int len, unsigned int flags);
```

Esta función send y recv se usan para sockets de flujo y socket de datagramas conectados. Puesto que los sockets de datagramas no están conectados a una máquina remota, antes de poder enviar el paquete debemos aportar la dirección de destino. A continuación se abordarán las funciones que se usan para sockets de datagramas desconectados.

Sendto: Se usa para enviar datos.

```
int sendto(int sockfd, const void *msg, int len, unsigned int flags, const struct sockaddr *to, int tolen);
```

Recvfrom: Se usa para recibir datos.

```
int recvfrom(int sockfd, void *buf, int len, unsigned int flags, struct sockaddr *from, int *fromlen);
```

De nuevo es igual, pero con dos elementos mas. Y por último la función que sirve para cerrar la conexión del descriptor de socket.

Close: Se usa para cerrar la conexión con el nodo remoto.

```
close(sockfd);
```

Existen muchísimas funciones que se usan en el manejo de socket, pero estas se las dejamos a usted para que las pueda investigar si le interesa el tema. Hasta este punto sólo se ha hablado de las herramientas que provee el sistema para programar socket, ahora se mostrará un pequeño ejemplo de un cliente-servidor con socket.

Servidor

```
#include <iostream>
using namespace std;
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#define PORT 3550 /* El puerto que será abierto */
int main()
{
    int fd, fd2; /* los ficheros descriptores */
    struct sockaddr_in server;
    struct sockaddr_in client;
    socklen_t sin_size;
    int BACKLOG=2; /* El número de conexiones permitidas */
    if ((fd=socket(AF_INET, SOCK_STREAM, 0)) == -1 ) {
        printf("error en socket()\n");
```



```
        exit(-1);
    }
    server.sin_family = AF_INET;
    server.sin_port = htons(PORT);
    server.sin_addr.s_addr = INADDR_ANY;
    bzero(&(server.sin_zero),8);

    if(bind(fd,(struct sockaddr*)&server, sizeof(struct sockaddr))== -1) {
        printf("error en bind() \n");
        exit(-1);
    }

    if(listen(fd,BACKLOG) == -1) {
        printf("error en listen()\n");
        exit(-1);
    }

    while(1) {
        sin_size=sizeof(struct sockaddr_in);
        if ((fd2 = accept(fd,(struct sockaddr *)&client, &sin_size))== -1) {
            printf("error en accept()\n");
            exit(-1);
        }
        printf("Se obtuvo una conexión desde %s\n", inet_ntoa(client.sin_addr) );
        send(fd2,"Bienvenido a mi servidor.\n",22,0);
        close(fd2);
    }
    return 0;
}
```

Cliente

```
#include <iostream>
using namespace std;
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#define PORT 3550
#define MAXDATASIZE 100

int main(int argc, char *argv[])
{
    int fd,numbytes; /*ficheros descriptores */
    char buf[MAXDATASIZE];
    struct hostent *he;
    struct sockaddr_in server;
    if (argc !=2) {
        printf("Uso: %s <Dirección IP>\n",argv[0]);
        exit(-1);
    }

    if ((he=gethostbyname(argv[1]))==NULL){
```



```
    printf("gethostbyname() error\n");
    exit(-1);
}

if ((fd=socket(AF_INET, SOCK_STREAM, 0))==-1){
    printf("socket() error\n");
    exit(-1);
}

server.sin_family = AF_INET;
server.sin_port = htons(PORT);
server.sin_addr = *((struct in_addr *)he->h_addr);
bzero(&(server.sin_zero),8);

if(connect(fd, (struct sockaddr *)&server,sizeof(struct sockaddr))==-1){
    printf("connect() error\n");
    exit(-1);
}

if ((numbytes=recv(fd,buf,MAXDATASIZE,0)) == -1){
    printf("Error en recv() \n");
    exit(-1);
}

buf[numbytes]='\0';
printf("Mensaje del Servidor: %s\n",buf);
close(fd);
}
```

Conclusiones

De manera general se han abordado algunos rasgos principales del tema de la programación de socket, los cuales resultan de suma importancia para aquellos que desean hacer que sus aplicaciones se comuniquen entre sí.



Encriptar el sistema de ficheros en Debian & Ubuntu

Eiger Mora Moredo, Yunier Vega Rodríguez
emora@estudiantes.uci.cu, yvrodriguez@estudiantes.uci.cu
Proyecto UNICORNIOS

A medida que se desarrollan herramientas, muchas de estas pueden utilizarse con objetivos diferentes para los que inicialmente fueron creadas; un vivo ejemplo es el potente Knoppix o cualquier otra compilación LiveCD de GNU/Linux, que permiten fácilmente a un tercero acceder a la información existente en una PC y modificarla a su antojo. La protección de datos sensibles tiene la prioridad siempre y cuando el entorno lo exija. El objetivo de este artículo es, dotar al lector de las herramientas necesarias para proteger su sistema mediante la encriptación.

Este artículo está basado en la configuración de Debian Etch con kernel linux-image-2.6.18-4-686. El primer paso es hacer todas las salvallas necesarias de los datos sensibles.

Instalación de sistema base

Lo ideal sería una instalación desde cero con el particionamiento más cómodo al usuario. No obstante, si están creadas las particiones de antemano no tiene problema alguno. El en caso de estudio se presenta el siguiente particionamiento:

boot	/dev/sda1	50MB
root_inicial	/dev/sda2	512MB
root_final	/dev/sda3	10GB
home	/dev/sda5	70GB

Se hará una instalación base ("sistema mínimo") en la que se definirá:

- La partición BOOT con el sistema de fichero deseado (ext3, reiserfs, xfs) que no se encriptará.
- La partición ROOT_INICIAL desde la que se harán las configuraciones y al final del día será la partición SWAP.
- Las particiones ROOT_FINAL y HOME que se marcarán como "no utilizar".
- Otras particiones que el usuario desee crear que se marcarán como "no utilizar".

Nota: No se define ninguna partición SWAP.

Encriptando

Instalado todo lo necesario, empezamos a encriptar las particiones ROOT_FINAL y HOME para esto se hace uso de la herramienta cryptsetup.

```
# cryptsetup -y -c aes-cbc-essiv:sha256 \ -h sha512 luksFormat /dev/sda3
# cryptsetup -y -c aes-cbc-essiv:sha256 \ -h sha512 luksFormat /dev/sda5
```

Para explicar las opciones:

-y verificar contraseña

-c especificación de Sistema de cifrado

-h hash a usar para la contraseña

luksFormat crea una partición LUKS y establece la contraseña inicial

El sistema le advertirá al usuario que los datos de la partición serán borrados y le pedirá que escriba YES como confirmación, seguido le pedirá la contraseña y la verificación de la misma.

```
# cryptsetup luksOpen /dev/sda3 root
# cryptsetup luksOpen /dev/sda5 home
```

luksOpen Abre la partición luks y establece el nombre de un nuevo dispositivo /dev/mapper/



root,home nombre del nuevo dispositivo: /dev/mapper/root & /dev/mapper/home respectivamente

Le pedirá la contraseña al usuario, el comando tendrá éxito luego de una correcta verificación. A continuación se da formato a los dispositivos creados y se montan:

```
# mkfs -t ext3 /dev/mapper/root
# mkfs -t ext3 /dev/mapper/home
# mount /dev/mapper/root /mnt
# mkdir /mnt/home
# mount /dev/mapper/home /mnt/home
```

Se copia el contenido de ROOT_INICIAL a la partición ROOT encriptada (/dev/mapper/root):

```
# cp -axv / /mnt
```

CHROOT: Configuraciones finales

El sistema base en /dev/mapper/root y /dev/mapper/home ya está encriptado pero es necesario configurarlo para que logre cargar, por esa razón el siguiente paso es un chroot. Se montan los dispositivos para acceder a ellos desde el chroot:

```
# mount -o bind /dev/ /mnt/dev
```

Se realiza en chroot y los montajes necesarios:

```
# chroot /mnt /bin/bash
# mount /boot
# mount /proc
# mount /sys
```

Se editan los siguientes ficheros de configuración:

```
/etc/fstab
/etc/kernel-img.conf
/etc/crypttab
/boot/grub/menu.lst
```

El /etc/fstab para estas particiones quedaría así:

<file system>	<mount point>	<type>	<options>	<dump>	<pass>
/dev/mapper/root	/	ext3	defaults	0	0
/dev/mapper/home	/home	ext3	defaults	0	0

En el fichero /etc/kernel-img.conf se agrega la siguiente línea:

```
ramdisk = /usr/sbin/mkinitramfs
```

En el /etc/crypttab se escribe lo siguiente:

<target name>	<source device>	<key file>	<options>
root	/dev/sda3	none	luks
home	/dev/sda5	none	luks

En el archivo /boot/grub/menu.lst se pone la línea:

```
# kopt=root=/dev/mapper/root devfs=mount ro
```

Nótese que el # se mantiene al inicio de la línea.

Se reconfigura el initrd.img del kernel:

```
# dpkg-reconfigure linux-2.6.18-4-686
```

Ahora, se desmonta todo, salimos de chroot y reiniciamos:

```
# umount -a
```



```
# exit  
# reboot
```

Encriptando la SWAP

Ahora se trabaja desde el sistema encriptado, pero falta la swap, para hacerlo seguimos los pasos anteriores:

```
# cryptsetup -y -c aes-cbc-essiv:sha256 \ -h sha512 luksFormat /dev/sda2  
# cryptsetup luksOpen /dev/sda2 swap  
# mkswap -v1 /dev/mapper/swap
```

Agregamos la línea correspondiente al fichero /etc/fstab:

```
# <file system>      <mount point>    <type>    <options>    <dump> <pass>  
/dev/mapper/swap     none             swap      sw           0      0
```

y ahora al fichero /etc/crypttab:

```
# <target name>    <source device>  <key file> <options>  
swap              /dev/sda2        none       luks
```

Ultimando detalles

Ya tenemos nuestro sistema de ficheros encriptado pero hay que poner la contraseña para cada partición en la carga del sistema, para evitar esto se pueden definir ficheros llaves para las particiones home y swap. Se hace el directorio destino(en el caso de estudio: cryptkeys)

```
# mkdir /etc/cryptkeys
```

Se genera el fichero llave:

```
# cryptsetup luksAddKey /dev/sda2 /etc/cipherkeys/swap  
# cryptsetup luksAddKey /dev/sda5 /etc/cipherkeys/home
```

Se le cambian los permisos a cada fichero para que solo root los pueda leer

```
# chmod 600 /etc/cryptkeys/*
```

Se modifican las líneas correspondientes en el fichero /etc/crypttab

```
# <target name> <source device>    <key file>    <options>  
home           /dev/sda5      /etc/cryptkeys/home    luks  
swap           /dev/sda2      /etc/cryptkeys/swap     luks
```

Se reconfigura el initrd.img

```
# dpkg-reconfigure linux-2.6.18-4-686
```

Se reinicia y listo, a instalar lo que le falta al desnudo sistema base inicial, para disfrutar de GNU/Linux sin temor a que los datos sensibles sean comprometidos.



El buscador, la solución contra el tiempo

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

La arquitectura y organización de la información son temas sumamente importantes cuando los volúmenes de ésta aumentan exponencialmente; los niveles investigativos actuales de las empresas e instituciones necesitan no sólo formas de organizar bien la información, sino herramientas para encontrarla fácilmente. Empresas como Google o Yahoo! brindan dichos servicios, pero para ello debemos de entregar nuestra documentación a ellos, y pagar además por el servicio.

El mundo del Software Libre ofrece alternativas, libres y además gratis que nos permiten indexar la información de nuestras intranets hospedada en un servidor Web, y encontrarla rápidamente.

Realmente sería imposible realizar una búsqueda de cualquier información a través de la red de redes; tal proceso se lleva a cabo de esta manera actualmente aquí en la Universidad de las Ciencias Informáticas (UCI) en el servidor de documentación libre montado desde febrero del año 2005, cuando se iniciaba el proceso de migración tecnológica de Software Privativo a Software Libre en la facultad 10.

A partir del montaje de este servidor y hasta nuestros días no existía un buscador, de ahí surge la tarea de estudiar varias aplicaciones de búsqueda que son capaces dada una pagina inicial y partir de una aplicación araña (spider) comenzar a realizar una búsqueda en un servidor Web indexar la información creando un buscador para agilizar los procesos de gestión de la información. Han sido estudiadas un total de 5 aplicaciones de las más usadas en internet y una quedó por estudiar debido a su complejidad de instalación y configuración, recomendamos para una institución profesional realizar un estudio sobre NUTCH, un buscador muy prometedor, las principales aplicaciones estudiadas se muestran a continuación:

Site Search Pro³²:

Descripción: Sistema comercial para añadir un motor de búsqueda a un sitio Web. Disponible tanto en ASP como en PHP. Dispone de varias versiones con funcionalidades más o menos avanzadas a precios distintos.

Licencia: Licencia comercial

Plataformas: Multiplataforma

Idiomas: Inglés

Experiencia personal: Este sistema no lo hemos instalado debido a que no lo poseemos, es comercial y su código no está accesible en la Web.

TSEP: The Search Engine Project³³:

Descripción: TSEP es un motor de búsqueda para un sitio Web, creado en PHP. Puedes poner un botón "Buscar en este Sitio" y permitir que la gente encuentre fácilmente lo que están buscando.

Licencia: Software Libre, GPL

Plataformas: Multiplataforma

Idiomas: Multidioma

Experiencia personal: Este sistema no pudo ser instalado satisfactoriamente debido a problemas en la implementación del mismo, se le fueron instalados todos los requerimientos que eran necesarios pero por errores de la implementación del mismo no funcionó correctamente.

PHPDig³⁴:

Descripción: Spider o robot y motor de búsqueda creado en PHP y con base de datos MySQL. Crea un glosario con palabras encontradas en las páginas indexadas.

³² <http://www.desarrolloWeb.com/scripts/site-search-pro-motor-busqueda.html>

³³ <http://www.desarrolloWeb.com/scripts/step-motor-busqueda-php.html>

³⁴ <http://www.desarrolloWeb.com/scripts/phpdig-motod-busqueda-php.html>



Licencia: Software Libre, GPL

Plataformas: Multiplataforma

Idiomas: Inglés

Experiencia personal: Sistema más trabajado, instalado correctamente e indexa correctamente los formatos html y txt , pero posee grandes problemas con la indexación de otros formatos y problemas en la base de datos.

Blasten blt-SEARCH³⁵:

Descripción: Un buscador adaptado a usted y sus visitantes. Utilizando un sistema escaneador de enlaces, basado en la tecnología PageRank con un poder de penetración superior a los 10 puntos, logra la búsqueda de imágenes.

Licencia: Freeware

Plataformas: multiplataforma

Idiomas: Español

Experiencia personal: Sistema instalado satisfactoriamente , mejor interfaz de todos lo que se han probado , libre, además en español , pero código muy confuso , a la hora de la indexación presenta un problema que impide este proceso, es muy configurable, tal vez con conocimientos de PHP se puedan solucionar los problemas que este posee.

Htdig³⁶:

Descripción: Sistema creado utilizando CGIs , son un conjunto de aplicaciones programadas en C++ , encargadas de buscar e indexar información de la Web , modular y posee complementos adicionales programados en lenguajes como C,C++ y Perl.

Licencia: Freeware

Plataformas: Multiplataforma

Idiomas: Multidioma

Experiencia personal: Sistema instalado satisfactoriamente , funcionalidades trabajando de manera total , probado 100% sobre el entorno GNU/Linux Debian 4.0 con la versión 3.1.6 de este producto compilada para esta arquitectura, sistema elegido para hacer el buscador del servidor de documentación libre de la UCI.

Htdig, el buscador seleccionado

Durante este proceso de investigación se ha seleccionado el Htdig debido a que se ajusta a nuestras características de una mejor manera, a continuación mostramos algunas ventajas a modo de características de este software:

- Es Software Libre completamente liberado bajo la GPL.
- Está programado en C++ utilizando CGIs.
- La indexación es mucho mas rápido.
- La búsqueda de datos es mucho menor que los otros que están programados en PHP.
- Es un sistema modular lo que le permite adicionar funcionalidades.
- Interfaz para las búsquedas configurable.
- Capaz de indexar muchos formatos, pdf, html, doc, txt, etc.

Este artículo ha sido el resultado de una investigación sobre los buscadores libres más importantes disponibles en internet, se han probado y estudiado las características de cada uno de ellos, quedando como constancia una guía completa de instalación del htdig³⁷.

³⁵ Blasten blt-SEARCH, <http://www.desarrolloWeb.com/scripts/blasten-motor-busqueda.html>

³⁶ Htdig, <http://www.desarrolloWeb.com/scripts/blasten-motor-busqueda.html>

³⁷ Yoandy Perez Villazon, Htdig, Indexando la red, 2007,
http://10.128.50.121/Documentacion/Publicaciones/15_Noviembre_2007/



Aplicación de la estrategia de migración a Software Libre

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

Este artículo es una recopilación de las principales aplicaciones de la metodología de migración hacia Software Libre creada en la UCI; identifica los lugares y eventos en el que ha sido expuesta dicha guía por los distintos miembros de la Comunidad de Software Libre de la UCI. La guía fue elaborada por el Ing. Ramón Paumier Samón como parte de su trabajo de curso del pasado año, y recopila la información de los avances del proyecto durante su tiempo de trabajo.

¿Que es la guía de migración ?

A finales del curso 2006-2007 y como resultado de una tesis de grado, que acumuló en su contenido todo lo aprendido por los integrantes del Proyecto de Servicios Especializados para la migración hacia Software Libre se lanzó la Estrategia de Migración hacia Software Libre de la UCI, la cual tuvo desde el mismo comienzo de su elaboración gran impacto en la soberanía tecnológica en los lugares donde fue aplicada.

Esta estrategia consiste en la acumulación de conocimiento de personas dedicadas a la noble tarea del estudio de las tecnologías libres con el afán de aplicarlas a nuestra universidad e instituciones que la necesitan para migrar a Software Libre, contiene un resumen de las experiencias de los principales intentos de migración de distintas universidades y otras organizaciones del mundo estudiando los principales éxitos y fracasos de estas para que no sean cometidas en el futuro, constituye una guía donde se estudian los principales procesos de la UCI y cómo se debe emprender la migración de los mismos de una manera que no existan grandes "traumas" en el desarrollo del cambio. Se describen los distintos pasos por los que se debe transitar para lograr una migración de manera eficiente proponiendo alternativas basadas en modelos.

Se exponen además los principales momentos por los que transcurre el proceso y qué hacer en cada momento, resaltando el apoyo que debe brindar la institución para el mismo, la formación de recursos humanos. También es descrita una propuesta en cuanto a las herramientas alternativas de código abierto libres a utilizar con la implantación de los nuevos Sistemas GNU/Linux, contiene un conjunto de plantillas necesarias para realizar un levantamiento de requisitos asociados a la tecnología previo a la migración.

Lugares donde ha sido aplicada la guía de migración

Aunque la guía de migración fue elaborada inicialmente para la UCI, ha sido aplicada en diversos lugares dentro y fuera de la universidad.

- Migración del grupo de multimedia de la UCI mediante la capacitación del personal y cambio del sistema operativo.
- Asesoría de migración y capacitación a la facultades 1, 2 y 7.
- Asesoría de migración a la facultades 4 y 6.
- Asesoría de migración al MES e instalación de varios servicios.
- Asesoría de migración a 1era y B (Oficinas adjuntas al Consejo de Estado) y capacitación.
- Asesoría de migración al centro Martín Luther King.
- Exposición del proceso y experiencias en reunión de trabajo con el grupo FORMATUR.
- Experiencias de la guía aplicándose al proyecto MINPPAL con Venezuela.
- Aplicación de la guía en la MiniUCI de Granma.
- Aplicación de la guía en la MiniUCI de Artemisa.

Esta guía de migración sirve de base a tesis de grado que se aplicarán en distintos lugares del país como empresas y escuelas a distintos niveles. Fue aplicada durante el proceso de formación de profesores de los distintos Institutos Politécnicos de Informática del país.



Referencias

Ing. Ramón Paumier Samón, Metodología para la Migración a Software Libre de la Universidad de las Ciencias Informáticas. UCI, Ciudad de la Habana, 2007

http://10.128.50.121/Documentacion/Publicaciones/09_Mayo_2007/

(Sólo disponible desde la UCI)

Marcos Luis Ortiz Valmaseda, Yoandy Pérez Villazón. Elementos para una Migración a Software Libre, Reunión de Migración de las escuelas de Formación del Turismo, 2006.