NÚMERO 3 AÑO 0

papirux

La Revista Libre

ISSN 2013-259X

al fin... Hemos regresado!!

Nhopkg... El ahora y el futuro

Apostar por el Software Libre

COMANDOS GNU/Linux (II)

Y el nombre es...

papirux
La Revista Libre

Índice

Equipo:

Coordinación:
Sergi Caparrós

Colaboradores:
Sergi Caparrós, SoLiDa Software_Libre_Dachary, Carlos Sánchez, martelj, Adrián Zavala Coria, Erko, Fausto Mauricio Lagos Suárez, Juan Carlos, Ricardo Rueda García, Cristiam Edwin Barreto

Diseño y Maquetación:
Pablo Axeitos

Correctores ortográficos:
Joferrue, Elmer, Fredy Enrique Lopez Pabon

Editorial

Ya tienes en tus manos Papirux nº 3, la primera publicación del año 2009.

Esta edición se ha hecho esperar un poco pero la espera ha merecido la pena.

Venimos con un aire totalmente renovado, un nuevo diseño de la revista más actual y fresco. También tenemos nuestra nueva mascota. ¿Qué aún no la conoces? ¿A qué esperas para saludarla?

A partir de este número incorporamos el número ISSN, un número identificativo internacional para revistas (tanto electrónicas como físicas).

Como puedes ver este número está repleto de cambios y novedades sin olvidarnos, por supuesto, del contenido de la revista. Esperamos que os gusten estos nuevos cambios y podamos aprender todos juntos nuevos temas y nuevas características del Software Libre.

Os animamos a todos aquellos que queréis colaborar con el Software Libre y en concreto con Papirux a que nos enviéis vuestros artículos a la dirección papirux@papirux.org y así podamos seguir avanzando y aprendiendo unos de otros.

Hasta el próximo número !!!

Y el nombre es...

Pablo Axeitos

TuxTanKamón!! La nueva mascota de Papirux ya cuenta con un nombre propio, elegido por una abrumadora mayoría. Agradecemos sinceramente vuestra participación en las votaciones de Papirux.org, ahora, y como siempre, este proyecto es un poco más vuestro.

Como veis TuxTanKamón viene acompañado de un montón de cambios (esperamos que os gusten) en el diseño de la revista, por desgracia esto también ha provocado un retraso en la publicación de este número (y os pedimos disculpas por ello).

El diseño ha sido realizado con Inkscape para los gráficos vectoriales, que son la mascota, el logotipo, cabeceras y piés de página... Para el retoque fotográfico he utilizado el casi-todopoderoso GIMP, pero el cambio más importante (y sufrido al principio, creedme) ha sido realizar la maquetación en un programa tan potente como Scribus. También he utilizado Fontmatrix para la selección de tipografías, que por cierto elimina todas las tipografías privativas de nuestra revista (decidle adiós para siempre a la Times y Arial de M$). La selección de colores, con el programa Agave, bastante práctico también, y la edición de textos con OpenOffice.

¿Por qué os cuento todo esto? Pues porque Papirux, además de vuestra revista, es una especie de reto personal para mí; es una forma de demostrar que se puede hacer diseño de calidad con las herramientas que el Software Libre ofrece. Os juro (de verdad) que en ningún momento he echado de menos ni el Corel, ni el Freehand, ni el Illustrator, ni siquiera el grandioso Photoshop... Es más, el coste de estas herramientas no nos permitiría llevar a cabo proyectos como este. Esto viene a demostrar las limitaciones que nos impone el software privativo "si no tienes dinero, no puedes publicar" (¿Os podéis imaginar a Cervantes sin dinero para pagar la licencia de una pluma?).

Es por esto que os animo a colaborar en desarrollo de aplicaciones de Software Libre, seréis los primeros en beneficiaros, pero no los únicos; si colaboráis, por ejemplo en el desarrollo de programas educativos, millones de personas en todo el mundo podrían salir beneficiadas de ello.

Quizás más adelante, cuando haya aprendido a usar estas GRANDES herramientas, me anime a hacer un tutorial, para que también vosotros les saquéis partido, pero mientras tanto me queda muchísimo que aprender...

Espero de verdad que os guste el nuevo formato de Papirux y nuestra (vuestra) nueva mascota TuxTanKamón, os puedo asegurar que para mí ha sido un auténtico placer el poder aportar algo en este mundillo que cada vez me tiene más enganchado.

Siempre vuestro (bajo licencia GPL3)
Pablo Axeitos

Software Libre Explicado

Por SoLiDa Software_Libre_Dachary

La comunidad de SoLiDa comparte con nosotros su experiencia en el software libre.

Sitio web: http://www.solida.dachary.edu.ar/

El “Software Libre'' es un asunto de libertad, no de precio. Para entender el concepto, debes pensar en “libre'' como en “libertad de expresión'', no como en “cerveza gratis'' (en inglés una misma palabra (free) significa tanto libre como gratis, lo que ha dado lugar a cierta confusión).

“Software Libre'' se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. De modo más preciso, se refiere a cuatro libertades de los usuarios del software:

La libertad de usar el programa, con cualquier propósito (libertad 0).

La libertad de estudiar cómo funciona el programa, y adaptarlo a tus necesidades (libertad 1). El acceso al código fuente es una condición previa para esto.

La libertad de distribuir copias, con lo que puedes ayudar a tu vecino (libertad 2).

La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie. (libertad 3). El acceso al código fuente es un requisito previo para esto.

Un programa es software libre si los usuarios tienen todas estas libertades. De esta manera deberías tener la libertad de distribuir copias, sea con o sin modificaciones, sea gratis o cobrando una cantidad por la distribución, a cualquiera y a cualquier lugar. El ser libre de hacer esto significa (entre otras cosas) que no tienes que pedir o pagar permisos.

También deberías tener la libertad de hacer modificaciones y utilizarlas de manera privada en tu trabajo u ocio, sin ni siquiera tener que anunciar que dichas modificaciones existen.

Si publicas tus cambios, no tienes por qué avisar a nadie en particular, ni de ninguna manera en particular.

La libertad para usar un programa significa la libertad para cualquier persona u organización de usarlo en cualquier tipo de sistema informático, para cualquier clase de trabajo, y sin tener obligación de comunicárselo al programador o a alguna otra entidad específica.

La libertad de distribuir copias debe incluir tanto las formas binarias o ejecutables del programa como su código fuente, sean versiones modificadas o sin modificar (distribuir programas de modo ejecutable es necesario para que los sistemas operativos libres sean fáciles de instalar). Está bien si no hay manera de producir un binario o ejecutable de un programa concreto (ya que algunos lenguajes no tienen esta capacidad), pero debes tener la libertad de distribuir estos formatos si encontraras o desarrollaras la manera de crearlos.

Para que las libertades de hacer modificaciones y de publicar versiones mejoradas tengan sentido, debes tener acceso al código fuente del programa. Por lo tanto, la posibilidad de acceder al código fuente es una condición necesaria para el software libre.

Nhopkg... El ahora y el futuro[0]

Carlos Sánchez (reeditado por martelj)

Una de las cosas que causa quebraderos de cabeza para un usuario nuevo en GNU/Linux[1], sobre todo si viene de MS Windows, son la forma de mantener y administrar los programas o paquetes, para solucionar esto se crearon los gestores de paquetes, entre ellos DPKG (Debian[2]) y RPM (Red Hat[3]) por mencionar algunos. Hoy quiero hablarles de uno en particular, que poco a poco se da a conocer en la red de redes, desde dos perspectivas el ahora y el futuro...

El Ahora...

Su Nombre es Nhopkg[4] el gestor de paquetes universal y está escrito en BASH[5], su autor Jotahacker[6], originalmente lo hizo para su sistema LFS[7] (Nhoax), lo consideraba vital para publicar su distro basada en LFS, finalmente terminó concentrándose en su desarrollo más que su sistema GNU/Linux[1] en sí. Su objetivo principal al crear Nhopkg, era su facilidad de uso y que fuese capaz de compilar, instalar y desinstalar paquetes, pero sin dejar de lado el contacto con los comandos de compilación e instalación básicos que se aprenden en un LFS[5].

Es importante destacar lo sencillo que es crear e instalar paquetes con Nhopkg, ya que permite al usuario hacer todo desde el mismo comando. A continuación unos ejemplos realizados con su última versión estable es la 0.4.2

Creando Paquetes desde las fuentes originales (.tbz,*.tgz,*.tar.gz,*.tar.bz2) (se usa la opción -c o --create-source)*

```
usuario [/sources] # sudo nhopkg -c paquete-ejemplo-0.1.0.tar.bz2
Preparando la creación del paquete ...
Por favor, introduzca el nombre del paquete. Ej.: wine
paquete-ejemplo
Introduzca la versión del paquete. Ej.: 1.0.1
0.1.0
Introduzca la revisión del paquete (No es necesária). Ej: 2
1
Es un paquete tar.bz2. Descomprimiendo ...
Introduzca los comandos de compilación. Finalmente presione Enter dos veces.
./configure --prefix=/usr
make

Introduzca los comandos de instalación. Finalmente prresione Enter dos veces.
make install

El nuevo paquete es /sources/paquete-ejemplo-0.1.0-1-src.nho
```

```
usuario [/sources] # sudo nhopkg -b paquete-ejemplo-0.1.0-1-src.nho
Preparando la instalación...
Descomprimiendo paquete-ejemplo-0.1.0-1-src.nho ...
Contruyendo paquete-ejemplo-0.1.0 ... Por favor, Espere ...

################################
# Todo el proceso de compilación...
################################

Instalando paquete-ejemplo-0.1.0 ... Por favor, Espere ...
Creando paquete-ejemplo-0.1.0-1.nho ... Ésto puede tomar algún tiempo ...
El nuevo paquete es /sources/paquete-ejemplo-0.1.0-1.nho
Limpiando...
```

Compilando y creando binario con el paquete (-src.nho) (se usa la opción -b o --build)

```
usuario [/sources] # sudo nhopkg -i paquete-ejemplo-0.1.0-1.nho
Preparando la instalación...
Descomprimiendo paquete-ejemplo-0.1.0-1.nho ...
Instalando paquete-ejemplo-0.1.0 ...
Por favor, Espere ...
Configurando su sistema ...
Limpiando...
```

Instalando un binario precompilado (.nho) (se usa la opción -i o --install)*

```
usuario [/sources] # sudo nhopkg -r paquete-ejemplo-0.1.0-1
Desinstalando paquete-ejemplo-0.1.0 ...
paquete-ejemplo-0.1.0 ha sido desinstalado.
Limpiando...
```

Removiendo un paquete (se usa la opción -r o --remove)

Nhopkg aún no cuenta con un gestor de dependencias, por lo que se debe tener en cuenta que existan en nuestro sistema las dependencias del paquete que deseamos construir...

El Futuro...

Nhopkg está en constaste desarrollo, por lo tanto, su próxima versión (la 0.5) programada para ser lanzada en el transcurso del primer trimestre del 2009 bajo la licencia GPL en su versión 3, vendrá con el esperado gestor de dependencias, - según indica la Web del Proyecto - ésta nueva característica lo hará mas sencillo para el usuario final, los paquetes construidos con la versión 0.5 tendrán más información sobre su contenido, como la descripción, licencia, arquitectura y más, también se ha reducido el tiempo que tardaba la versión 0.4 en generar los binarios.

Viendo aún más allá, su autor nos dice algo aún mas interesante:

Cito de su Blog:

"[...]aunque todavía no está decidido si entrará en esta última versión (0.5), estoy trabajando en una opción (en realidad serán varias) para que desde el mismo Nhopkg se puedan transformar paquetes .deb, .rpm y .tgz (usados por otros gestores de paquetes) a los paquetes .nho sin necesidad de tener instalado ni dpkg, ni rpm, ni, en definitiva, sus correspondientes gestores de paquetes. Ésto ya lo hace alien[8] pero para ello necesita de rpm y dpkg. Por lo cual, esta nueva cualidad de Nhopkg será importante."
Es importante notar que Nhopkg que aunque fue hecho para funcionar en un sistema LFS está hecho para ser universal, por lo tanto funcionará en cualquier distribución GNU/Linux.

Enlaces de interés:

- *0: http://el-simpicuitico.blogspot.com/2009/01/nhopkg-el-ahora-y-el-futuro.html*
- *1: http://es.wikipedia.org/wiki/GNU/Linux*
- *2: http://es.debian.org*
- *3: http://redhat.es*
- *4: http://nhopkg.sourceforge.net/*
- *5: http://es.wikipedia.org/wiki/Bash*
- *6: http://jotahacker.es/*
- *7: http://linuxfromscratch.org*
- *8: http://kitenet.net/~joey/code/alien/*
- *9: http://foro.jotahacker.es*
- *10:http://es.wikipedia.org/wiki/Nhopkg*

Apostar por el software libre

Adrián Zavala Coria

Hoy en todo el mundo muchas carreras de niveles superiores (sin importar la ciencia)se apoyan en la informática como una forma de ahorrar tiempo en sus procesos de investigación, desarrollo, calculo, etc.

El año pasado Richard Stallman habló mucho sobre la importancia que debe tener el software libre en la educación básica (primaria y secundaria). También muchos sabemos del fracaso que ha resultado "One Laptop per Child" por el simple hecho que Microsoft ha metido sus manos en un proyecto que todavía tiene buenas intenciones, pero se va corrompiendo al tener un sistema operativo propietario de por medio.

Pero olvidándonos un momento de OLPC, las personas entre 18 y 25 años que asistimos a una universidad (o instituto tecnológico) al parecer no se han dado cuenta de la dimensión que posee vivir todos los días con software propietario en el campus. Al parecer tener software propietario en los servidores, laboratorios y oficinas es lo más normal que existe en el mundo.

Si nos hemos dormido o simplemente no ha pasado por nuestra mente, deseo plantear la pregunta: ¿es bueno el software libre en las universidades?, porque hay fuertes argumentos como para tomarlo como una fuerte solución a muchos problemas en los campus.

Es cierto, GNU/Linux no es la solución para todos y no vale la pena tratar de abordar ese tema en este escrito. Lo que nos lleva a otro punto central: uso del software libre, pero en la búsqueda de nuevas tecnologías que sean beneficiosas a largo plazo.

Echemos un vistazo a las carreras del ramo de la construcción. Si hay una marca con la que se debe asociar a un ingeniero o arquitecto es: Autodesk, la empresa que desarrolla software CAD para este ramo. Si no manejas al menos un paquete de software de esta compañía, simplemente es difícil que obtengas trabajo en una constructora.

¿Y eso que tiene de malo? -muchos dirán- el software es de una calidad asombrosa y cumple muy bien con su función. Si todos usan los productos de esta compañía, es lógico que en las universidades se deba de formar al profesional con esa importante herramienta.

No tengo nada contra Autodesk, pero si pensaste lo anterior, es porque no has visto la verdadera esencia del problema. Y esta es muy simple: estamos casando al futuro profesional con una marca. Como dice Stallman: "es como enseñarle a fumar a un niño en un mundo en donde solo hay un [proveedor] de cigarros", con el tiempo será adicto y no habrá retorno.

Quizás suene demente y poco viable, pero el software libre aquí puede aparecer como una viable apuesta a futuro. Solo les mencionaré a grandes rasgos la forma que en el caso de México, se puede abordar el problema anterior.

Tendríamos que establecer un proyecto formal por parte del gobierno y la iniciativa privada. Todo esto para organizar a un grupo de expertos en el área teórica (ingenieros y arquitectos) que ayuden a los programadores en la formación del nuevo software. Estos últimos podrían ser contratados para el proyecto y el personal humano restante puede ser tomado de los mejores alumnos en el área de programación de las universidades. Al final, todo desarrollado bajo los términos de una licencia y una plataforma GPL.

Sí, es descabellado pensar en algo así. "Regalar" el producto final a cualquiera, Malgastar el dinero para que otro lo use sin invertir nada. Pero insisto, si piensas de la forma anterior, estás viendo de nuevo la superficie; no del problema, sino de la solución.

La GPL daría beneficios a largo plazo inimaginables. Cada universidad y constructora obtendría la liberación del pago de licencias por el software, se descubrirían nuevos talentos en muchas áreas, se impulsaría la investigación y desarrollo de tecnología, se trabajaría bajo una "norma" en común, todo para dar paso a una más importante: la naturaleza libre de la GPL protegería la inversión ante posibles modificaciones y peligros.

Imaginemos que México crea el software, este llega a España y ellos lo mejoran. México se beneficia de lo hecho en España. Ahora lo toman los argentinos e incrementan las funciones del software. Argentina y España se benefician con la inversión que México inicio, pero este obtiene también las mejoras. Y podemos hacer las combinaciones que queramos, al final todos nos beneficiamos, nadie pierde, solo Autodesk.

No haré más largo el texto. Si has llegado hasta acá es porque te he puesto a pensar y has entendido el punto: el software libre es la mejor opción a la que puede recurrir la educación para buscar nuevos conocimientos y tecnologías. A partir del ejemplo mencionado, tú también puedes empezar a pensar en otros ejemplos. Solo es cuestión de pensar bien.

Y tú... ¿de qué sabor prefieres?

Erko

¿A cuántos de ustedes, al ver la interfaz de su computadora, no les han preguntado que "Windows" usan..?, en el mejor de los casos pensarán que han gastado un buen dinero y les preguntarán por la versión de Mac OS que tienen...., creo que a la mayoría de ustedes ¿no...?. Y no es para menos, a mi me pasó lo mismo cuando vi los impresionantes efectos de Beryl, o los wallpapers animados de Enlightenment, o los efectos "3D" de FVWM-Crystal. Y yo no he visto ni un solo escritorio de Microsoft o Mac que haya sido configurado al mismo grado que un entorno GNU/Linux, si a caso podrán cambiar el fondo de escritorio, el color de los decoradores de ventanas y posiblemente los iconos, pero hasta ahí... y no por que no se pueda o no exista, si no por que o es "ilegal" o hay que "soltar un varo".

Sin duda una de las grandes ventajas de usar un sistema operativo basado en Unix (ya sea GNU/Linux, etc...) es la gran variedad de entornos gráficos (ya sean entornos de escritorio (DE), gestores de ventanas (WM) o inclusive una interfaz de sólo texto...) que tenemos a disposición para poder hacer pleno uso de ellos y crear ambientes totalmente personalizados y únicos. Y si sólo crees que con Compiz Fusion es posible tener un escritorio que haga que a mas de uno se le salgan los ojos estás muy equivocado y por eso hemos creado esta sección.

Aquí trataremos de presentarles la gran diversidad de entornos gráficos que existen en el mundo de GNU/Linux y lo que es posible lograr con estos.

Primero que nada hay dos grandes vertientes de interfaces gráficas en el mundo del software libre, los pequeños diablillos Gestores de Ventanas o Windows Managers y los monstruos Ambientes de Escritorio o Desktop Enviroment.

Quizás más de uno ya habrá oído o conoce la diferencia entre uno u otro, pero para los que no aquí tienen una pequeña introducción.

Ambas interfaces, tanto los gestores de ventanas, como los entornos de escritorio funcionan bajo el sistema X Window. Este paquete es el encargado de ejecutar varias aplicaciones en una misma pantalla gráfica y mostrar las ventanas que estas ocupan, además es el actual estándar que se utiliza para los entornos gráficos de GNU/Linux.

El gestor de ventanas corre bajo un ambiente X Window y es el encargado de controlar las ventanas que crea una aplicación, con el podemos mover las ventanas por la pantalla, es el que dibuja los botones de minimizar, maximizar y cerrar de las ventanas, dibuja los marcos de las ventanas entre otras cosas. La mayoría de los gestores ofrecen las mismas características y funciones, en lo que se diferencian es en la forma de manejar sus procesos y en la configuración de los mismos.

Después se encuentran los entornos de escritorio, estos son un conjunto de aplicaciones que se encargan de facilitar su configuración y de agregar una mayor funcionalidad al entorno gráfico. Digamos que un escritorio se compone básicamente de:

- *Un gestor de ventanas*
- *Un explorador de ficheros*
- *Un creador de lanzadores*
- *Una aplicación con la que podamos configurar la apariencia de nuestro escritorio*
- *Compatibilidad con applets*

Los gestores son muy veloces y ágiles, son muy recomendados en el caso de que tu PC sea antigua o de que requieras que todo el poder de procesamiento sea dedicado a tus aplicaciones, así que si tienen una PC que se este llenando de polvo te recomiendo que le des nueva vida con uno de estos entornos.

Los escritorios son más robustos, al componerse de diferentes aplicaciones, por lo que son un tanto mas lentos, aunque existen escritorios como XFCE que tratan de equilibrar la velocidad de procesamiento con la fácil configuración de un entorno.

Espero les agrade esta sección, así que comenzamos...

Ciencia de la mano del Software Libre

Lic. Fausto Mauricio Lagos Suárez

En el mundo de quienes trabajamos en ciencias como la matemática nos hemos encontrado con que las herramientas de software nos brindan un sin número de posibilidades a la hora de agilizar nuestro trabajo, presentar mejor los resultados de una investigación o simplemente motivar a nuestros discípulos a que estudien lo ladrilludo de nuestras ciencias; sin embargo parece ser que normalmente nos hemos acostumbrado a limitarnos a lo "malo conocido" y no queremos ver lo bueno por conocer, entonces bueno, esta colección de artículos esta dirigida precisamente a todos aquellos entusiastas de la ciencia (docentes, estudiantes, investigadores, etc) que aun no han entrado en el mundo del software libre para que se encuentren con todo aquello que siempre hemos soñado, aplicaciones libres de costos y con altísimas prestaciones, para todos ellos están las comparativas entre lo mejor del software libre y las aplicaciones de costo más generalizadas (cuidado que no se trata de ninguna guerra), también, obviamente, para todos aquellos que ya hacen parte del mundo del software libre pero que aún no han podido dar el salto definitivo pensando que no encontrarán el potencial de sus aplicaciones de costo, en las aplicaciones libres están los tutoriales de introducción a lo mejor de estas aplicaciones que definitivamente nada tienen que envidiarles a sus camaradas de costo.

Ahora sin más presentación la primera serie de esta colección está dedicada a la geometría, las aplicaciones disponibles para desarrollar construcciones geométricas que permitan ver propiedades, resaltar resultados, comprobar teoremas y muchos aspecto más del estudio y la investigación en geometría.

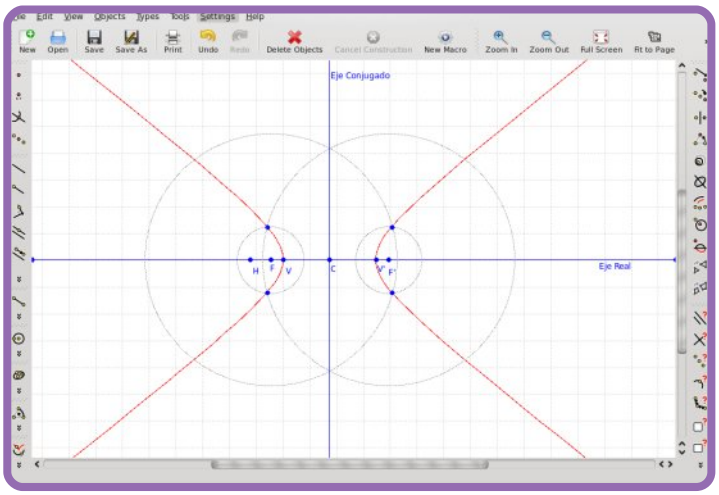
Las aplicaciones de Geometría Dinámica, como se les denomina por sus prestaciones de actuación dinámica sobre las construcciones, demuestran un gran potencial a la hora de ofrecer recursos para el estudio de la geometría, en nuestra primera comparativa tenemos a Cabri el gigante de la geometría dinámica en plataformas PC – Windows y Kig el más acertado (bajo mi experiencia) candidato a ser nuestra elección al saltar a Linux.

COMPARATIVA

Ninguno de quienes trabajamos en el entorno de las aplicaciones de Geometría Dinámica podemos desconocer al gigante Cabri, software que ha crecido considerablemente adoptando incluso herramientas de exportación a web a través de archivos Java, sin embargo a la hora entrar en Linux no se queda atrás el potencial de KIG llegando a ser la aplicación que llena a la perfección las botas del Cabri.

Esta comparativa esta hecha en 5 aspectos: Interfaz, Construcciones incorporadas, Ecuaciones y Coordenadas, Lugares Geométricos y Macros.

Interfaz



Respecto a la interfaz de usuario KIG, a mi parecer, aventaja a Cabri gracias a su presentación de las herramientas de construcción organizadas en barras laterales que facilitan y agilizan el acceso, además desde el menú contextual se encuentran opciones particulares de construcción, por ejemplo sobre una circunferencia se pueden encontrar opciones como tangente o centro de curvatura entre otras. Algo que completa la interfaz es la capacidad de exportar archivos, capacidad que si bien esta en construcción en el Kig, no es nada despreciable al incluir formatos como el del mismo Cabri, sin embargo Cabri incluye formatos de exportación como Java para publicación de las construcción en la Web.

Construcciones Incorporadas

En cuanto a las construcciones que incorpora por defeco Kig presenta unas opciones adicionales a las que tiene Cabri, opciones como la construcción de diferencia de vectores, iteración de transformaciones geométricas, tangentes a cónicas o circunferencias, centro y ejes de una cónica, Elipses e hipérbola por focos y un punto y otras opciones más que lo destacan.

Ecuaciones y Coordenadas

Respecto al manejo de gráficas de funciones y presentación coordenadas relativas a un punto sobre la curva resultante los dos, tanto Kig como Cabri ofrecen un soporte adecuado y suficiente para quienes además de las construcciones geométricas quieran involucrar en su trabajo la representación de ecuaciones.

Lugares Geométricos

La característica de la construcción de Lugares Geométricos hace que los procesadores geométricos tengan una excelente ventaja frente a cualquier otro tipo de software de graficación, al respecto Kig presenta la construcción y traza de lugares geométricos tan bien como se podría esperar de una aplicación de su nivel de prestaciones, sin embargo estamos a la espera de que en próximas versión se permita la selección de puntos sobre un lugar y la definición de intersecciones entre los lugares y otros elementos de la construcción.

Macros

Para quienes hayan hecho el salto o estén buscando hacer el salto a Linux, Kig ofrece un excelente soporte en cuanto a la geometría dinámica, muestra de ellos además de las características que ya se han mencionado es la construcción de Macros que no presente mayor diferencia al manejo de la misma herramienta en su competidor para Windows, Cabri, Kig maneja la construcción de macros con el mismo principio que Cabri y cuenta además con la opción de guardarlos como archivos independientes para tenerlos disponibles en otras sesiones de trabajo, muestra de ello es su integración como nueva herramienta en los menús de construcción, también permite la exportación de macros en varios formatos brindando así nuevas características que potencian el trabajo con esta herramienta.

Sin más en este primer artículo de la serie hemos hecho una presentación del potencial del KIG y queda claro que el salto a Linux no presenta traumatismos al encontrar las aplicaciones con el potencial de aquellas que estamos acostumbrados a manejar, queda ya en cada uno de los lectores la determinación de la conveniencia o inconveniencia del cambio, tal vez para ayudarlos a decidir quedamos a la espera, en la próxima entrega, del primero tutorial de introducción a la geometría dinámica con Kig.

CABRI
Versión 1.2.5
Autores: CabriLog SAS
Licencia: Comercial
S.O. Windows

KIG
Versión 1.0
Autores: D, Devriese, M. Paoline, P. Toscazo y F. Pasquarelli
Licencia GNU
S.O. Linux

Comentarios y sugerencias.
Lic. Fausto Mauricio Lagos Suárez
pirataxoo7@gmail.com

Comandos en GNU/Linux (II)

Sergi Caparrós

man: Las distribuciones Linux incluyen documentación de casi todos los programas, comandos y archivos instalados en el disco duro. El comando man se usa para ver la ayuda en línea o la documentación denominada páginas de manual. Para aprender sobre el comando man se deberá utilizar:

man man

xman: Si quieres utilizar la ayuda pero en tu entorno gráfico, puedes utilizar el comando xman. Se trata del manual de documentación pero en una ventana gráfica.

chfn: Sirve para cambiar la información Finger. Escribimos "chfn nombre_usuario" y nos solicitará el nombre real, departamento (habitación), teléfono oficina, de casa y otro. Esta información no modifica el nombre de usuario ni la contraseña. El comando chfn debemos ejecutarlo con permisos de root (sudo). Si hubiera información anterior en estos campos, aparecerá dentro de corchetes y pulsando ‹INTRO› sin escribir nada, no modificaremos el dato.

shutdown: Para reiniciar el sistema podemos utilizar el comando shutdownm, que se encuentra en /sbin (por lo tanto requiere permisos root para ejecutarlo). El comando shutdown tiene varias opciones. Por ejemplo, si se usa la opción -r o reboot, seguida por seguidala palabra now se reiniciará el sistema de manera inmediata: "shutdown -r now"

halt: Para apagar completamente el equipo podemos utilizar el comando halt. Se requiere permisos de root. "sudo halt"

ifconfig: Muestra las interfaces de red del sistema. Muy práctico para ver la configuración de red por cable.

iwconfig: Muestra los adaptadores de red inalámbricos y su información, como por ejemplo la velocidad y la red a l que está conectado. Muy práctico para ver la configuración de conexiones WIFI.

lspci: Lista los buses PCI y los dispositivos conectados a ellos (tarjetas de red, audio, vídeo, etc).

lsusb: Lista los buses USB y los dispositivos conectados a ellos como impresoras, wifis, pendrives, etc).

lshal: Lista todos los dispositivos detectados por HAL (Hardware Abstraction Layer, capa de abstracción del hardware).

lshw: Lista el hardware del equipo, incluyendo datos como fabricante, tipo de hardware y dónde está conectado.

cat: Abreviatura de concatenar. Sirve para ver el contenido de ficheros de texto (cat nombre_fichero.txt). También sirve para copiar un fichero de texto a otro, para eso escribiremos cat fichero1.txt › fichero2.txt

passwd: Sirve para cambiar la contraseña de los usuarios. Si ejecutas el comando sin ningún argumento, cambiarás la contraseña del usuario actual (siempre que conozcas la contraseña actual). Para cambiar la contraseña de cualquier usuario deberás escribir "sudo passwd nombre_de_usuario", no hace falta saber la contraseña del usuario pero sí la de root.

su: Para cambiar de usuario durante una sesión de terminal. Si quisieras validarte con el usuario pepe deberás utilizar "su pepe" y seguidamente escribir la contraseña del usuario pepe. Muy útil cuando estás en la sesión de un usuario sin permisos y necesitas ejecutar algún comando con permisos de root (o de cualquier otro usuario).

Alt+Ctrl+Fx: Sirve para cambiar de terminal. Tienes seis terminales y el séptimo que es donde se inicia el entorno gráfico. Para entrar al tty1 pulsar Alt+Ctrl+F1, para entrar en tty2 pulsar Alt+Ctrl+F2 y así sucesivamente hasta la tt6 que es la "terminal gráfica".

logout: Cierra la sesión del usuario en la terminal.

whatis: Si no estás seguro de lo que hace un programa determinado se puede utilizar el comando whatis. Para saber lo que hace el programa ls deberemos utilizar "whatis ls" y como resultado obtendremos una breve descripción de su utilidad.

apropos: Usa la base de datos de whatis para visualizar todo lo relacionado con un comando. Por ejemplo para saber en todas las páginas en las que aparece "ls" utilizaremos "apropos ls" y nos mostrará allí donde aparece "ls".

Uso de la tecla TAB: Una característica del intérprete es la terminación de los comandos. Si tecleas las primeras letras del nombre del nombre del comando y luego pulsas la tecla ‹TAB› aparecerá el comando escrito al completo. Si sólo se pulsan unas pocas letras tendrás que pulsar dos veces ‹TAB› y mostrará todos comandos que comiencen por las letras introducidas.

Programas en segundo plano (&): A veces resulta útil mandar programas a trabajar al segundo plano mientras nosotros seguimos trabajando con la terminal. Por ejemplo si tenemos un archivo muy grande que queremos comprimir o descomprimir. Este proceso tardará bastante tiempo, dejándonos el terminal sin poder utilizar hasta que no acabe la tarea. Para evitar que el programa esté ocupado en primer plano utilizaremos "&" al final para indicar que se ejecutará en segundo plano. Por ejemplo para descomprimir un fichero podríamos emplear "unrar -e nombre_fichero.rar &". Si llama al programa gedit comprobarás que el terminal queda ocupado hasta que el programa gedit se cierre. Por el contrario si ejecutas "gedit &" el programa gedit se estará ejecutando en segundo plano y mientras tanto podrás seguir utilizando el terminal sin necesidad de abrir otro terminal nuevo.

Personalización del GRUB

Juan Carlos

Grub es la herramienta encargada de iniciar los distintos sistemas operativos instalados en un ordenador.

Cuando se enciende el ordenador es el gestor de arranque el programa que se carga para que el usuario seleccione qué sistema operativo se iniciará.

Los dos gestores de arranque mas utilizados son GRUB y LILO. El primero es el más utilizado hoy en día, mientras que LILO sólo se encuentra en distribuciones ya obsoletas o como alternativa a GRUB.

PROCESO DE INICIO DE UN GESTOR DE ARRANQUE

La BIOS es carga en la memoria principal comprobando el estado del equipo con el proceso POST. En caso de error el inicio se detiene mostrando el problema detectado.

Al finalizar la ejecución de la BIOS se procede a la carga del MBR del primer dispositivo de almacenamiento.

El gestor de arranque toma el control, se suceden tres fases:

Se cargan los primeros 512 bytes que constituyen el MBR, al tener un tamaño tan pequeño, en esta fase se limita a cargar la siguiente etapa de arranque.

Muestra el menú con los sistemas operativos y núcleos disponibles que se pueden iniciar. Se requiere la intervención del usuario para seleccionar una opción o esperar el tiempo marcado para la carga automática de la opción por defecto.

Carga el núcleo y le pasa el control del arranque, este núcleo es cargado en “initrd”, un sistema de archivos temporal utilizado como medio de destino previo al acceso al verdadero sistema de archivos.

CARACTERÍSTICA DE GRUB

Las múltiples características integrada en GRUB lo han convertido en el gestor de arranque más destacado, por encima de LILO y de otras herramientas similares, tanto en Linux como en otros sistemas operativos:

Algunas de estas características son:

Configuración dinámica, es decir permite cambiar las opciones de arranque modificando un archivo sin necesidad de reinstalar GRUB en el MBR o directamente desde el menú de elección del sistema operativo.

Compatible con sistemas de archivos más utilizados, como ext2, ext3, ReiserFS FAT32 y NTFS

Totalmente personalizable, ya sea mediante imágenes o variando el color del menú.

CONFIGURACIÓN

En el directorio /boot/grub encontramos el archivo menú.lst encargado de modificar la configuración y opciones de arranque de GRUB

De las opciones disponibles las más importantes son:

Default 0: número de núcleo o sistema operativo a arrancar. Este valor está determinado por el orden en que los núcleos y sistemas operativos estén escritos en este mismo archivo, siendo 0 el primero.

Timeout 5: tiempo en segundos en que la opción de arranque predeterminada arranca si no se ha pulsado en el menú de GRUB tecla alguna.

Howmany=all: al reemplazar all por un número entero limita las opciones que aparecen en el menú. De especial utilidad si existen muchos núcleos instalados y preferimos no borrarlos.

Justo después de todos estos parámetros están las secciones con las diferentes opciones de arranque a mostrar en el menú GRUB. Tanto las distintas versiones del núcleo Linux como de otros sitemas operativos.

La plantilla que se debe utilizar para crear una nueva entrada de arranque en el menú de GRUB es:

```
Title Debian GNU/Linux,
kernel 2.6.18-6-686
root (hdo,0)
kernel /boot/vmlinux-2.6.18-6-686
root=/dev/sda1 ro
initrd /boot/initrd.img-2.6.18-6-686
```

Con las siguientes opciones: title, título a mostrar en el menú de GRUB, root, disco duro y número de partición a utilizar para arrancar; kernel, ruta al núcleo; e initrd, ruta al fichero con el sistema de archivos virtual.

CAMBIOS EN EL ARRANQUE

En el menú de GRUB, donde se muestran las distintas opciones que se pueden arrancar en la máquina, podemos modificar las opciones de carga de un núcleo. Por ejemplo para arreglar una ruta errónea o añadir nuevos parámetros en su inicio.

Para esto, en el menú GRUB nos situamos sobre la opción que queremos cambiar y presionamos la tecla "e". Se mostrarán las opciones de inicio del núcleo elegido, donde ponemos añadir una nueva línea pulsando "o", eliminar una, "d", o modificar una ya existente,"e". Para arrancar con los nuevos parámetros presionamos "b".

Estos cambios son temporales y se pierden en el siguiente arranque. Para conservarlos es necesario añadirlos al archivo menú.lst.

CAMBIOS VISUALES

COLORES

Para cambiar el color del menú donde elegimos el núcleo a iniciar, modificamos el archivo menú.lst del directorio de configuración de GRUB la opción color.

En esta opción indicamos los colores a utilizar por las opciones del menú y su fondo, cambian según estén o no seleccionadas.

Su sintaxis es: color cyan/blue White/blue, donde reemplazamos los nombres de los colores en inglés por los preferidos.

FONDO DE PANTALLA

Si queremos añadir una imagen de fondo al menú GRUB, hacemos uso de la opción splashimage indicando al menos una imagen en formato XPM.

Por ejemplo podemos descargar imágenes incluidas en Debian ejecutando: aptitude install grub-splashimages debían-edu-artwork-usplash

Las imágenes, se guardan en formato XPM, en el directorio /boot/grub/splashimages, comprimidas con gzip.

Es necesario indicar a GRUB la imagen que queremos utilizar en el arranque. Por ejemplo, para añadir la de nombre bike_gua.xpm.gz, insertamos en el archivo menú.lst, justo antes de las opciones relacionadas con los núcleos y sistemas operativos, la línea:

Splashimage=(hdo,o)/boot/grub/splashimages/debsplash.xpm.gz

Sustituyendo hdo,o por la unidad y partición donde esté el directorio /boot.

También podemos crear una imagen propia utilizando The Gimp, u otro programa para crear gráficos, pero deberemos tener en cuenta una serie de caracteristicas que debe cumplir esta imagen para insertarla en el GRUB.

Para crear una imagen para GRUB seguiremos los siguientes pasos:

Abrimos la imagen desde Archivo\abrir.

La transformamos a la resolución 640x480. Para lo que presionamos Imagen, escalar la imagen, donde asignamos los valores antes mencionados para los campos anchura y altura respectivamente.

Presionamos en Imagen\Modo\Indexado. En la ventana abierta activamos la opción generar paleta óptica y en Número máximo de colores escribimos 14 en sustitución del valor predeterminado 256.

La imagen generada la guardamos en formato Imagen X PixMap, con extensión XPM, y la comprimimos ejecutando: gzip imagen.xpm

Obtenemos un nuevo archivo con nombre imagen.xpm.gz que podemos añadir igual que hicimos con la opción splashimage para el fichero bike_gua.xpm.gz.

Sergi Caparrós

Hace poco preguntaba un usuario cómo instalar Frozen Bubble en algún foro. Si trabajas con una distribución basada en Debian, como Ubuntu, es tan sencillo como escribir en el terminal:

sudo apt-get install frozen-bubble

Cuidado al probar este juego porque es enormemente adictivo. Está basado en Puzzle Bubble. Frozen Bubble trabaja con licencia GPL por lo que es totalmente libre y está disponible para GNU/Linux, Windows y MacOS. Está programado con el lenguaje de programación Perl. La página oficial del juego es **www.frozen-bubble.org**.

Podemos arrancar el juego desde: "Aplicaciones › Juegos › Frozen-Bubble"

En el menú principal encontramos:

Start 1P Game
Juego para un jugador. Utiliza las flechas para jugar.

Start 2 Game
Juego para dos jugadores.
El jugador 1 utiliza las flechas:

El jugador 2 utiliza las teclas:

Start Lan Game
Para jugar una partida en una LAN (red local).

Start Net Game
Para jugar una partida a través de Internet (existen varios servidores).

Level Editor
Podrás editar tus propios niveles, sin ningún tipo de limitación. Sólo tu imaginación será responsable de los niveles creados. Para jugarlos deberás elegir la primera opción del menú (Start 1P Game) y la opción "Pick Level set and Start Level" del submenú.

Graphics
Alterna entre los tres modos gráficos posibles (pobre, normal y completo).

Change Keys
Podrás reconfigurar las teclas.

High Scores
Mostrará las mejores puntuaciones obtenidas.

Es posible arrancar el juego desde consola tecleando:

frozen-bubble

Si queréis ver de todas las opciones que dispone al arrancar desde consola podéis llamarlo con el parámetro – help.
frozen-bubble – help
fullscreen: Ejecuta en ventana completa
no-sound: Sin sonido
no-music: Sin música
slow-machine: para jugar en máquinas antiguas
very-slow-machine: para jugar en máquinas muy antiguas
solo: juega un solo jugador
no-time-limit: Desactiva el tiempo, ideal para niños pequeños.
color-blind: Las bolas contienen unos dibujos. Ideal para personas con problemas de daltonismo.

Desarrolladores:
Artwork: Alexis Younes
Amaury Amblard-Ladurantie
Soundtrack: Matthias Le Bidan
Design & Programming: Guillaume Cottenceau
Level Editor: Kim and David Joham

Una taza de café

Ricardo Rueda García

Hola compañero lector, quiero compartir mi experiencia con el software libre.

Hace varios años inicié en la informática con MS-DOS y Windows 3.1, igual que mucha gente pensaba que estos sistemas operativos eran la única opción, y aunque con algunos problemas estaba resignado a continuar con el sistema, así que pasé por las versiones 95, 98 y agregados hasta XP, con el que las cosas no estaban tan mal. Los problemas comenzaron hace un año cuando compré mi PC nueva, que ya tenía "de cajón" Windows Vista (es difícil encontrar otra opción en una PC nueva). Desde el principio noté un sistema lento pero aceptable (recuerden que yo vivía resignado), lo peor fue cuando quise instalar mis programas anteriores y me encontré con que no eran compatibles mostrándome infinidad de ventanas de error de instalación, ya saben: error XXX0033311 el sistema no pudo... el sistema no encontró... el sistema no responde... Bien, entonces tenía que pagar licencias nuevas para hacer lo mismo que hacía con mis programas "viejitos" sólo porque los archivos ya se guardaban con una extensión nueva.

Ya conocía algunos programas "gratuitos" y comencé a buscar, encontré Openoffice lo descargué, lo instalé y listo, así fácil, rápido y funcionó de maravilla... "El software libre me cautivó" seguí buscando y encontré soluciones para todas mis necesidades, todos trabajando muy bien, pero faltaba el corazón de la máquina, lo que hace funcionar al ordenador: el sistema operativo, ya había escuchado de Linux pero siempre lo consideré un sistema para "Geeks" aquellos que modifican el kernel y que desarrollan sus propias aplicaciones, pero afortunadamente para mi conocí Ubuntu (finalmente basado en Linux) y la verdad me atrapó, fue así que decidí cambiar y actualmente estoy muy contento con mi "nuevo" sistema operativo.

Obtener los programas fue fácil: me conecté a la Internet y listo, pero yo me pregunto: ¿Cómo era hace diez años? Se imaginan que posibilidades había para personas como yo que aunque tengo conocimientos en informática no soy un especialista, ahora con el movimiento del software libre que desde hace algún tiempo ha crecido es fácil conseguir aplicaciones de calidad sin costo para el usuario final, pero veamos el otro lado, sabemos que el software no se ha creado solo, a un grupo de personas les ha costado tiempo y mucho esfuerzo.

Para el trabajo y la diversión yo utilizo programas diversos: procesadores de texto, editores gráficos y de audio; aveces me pregunto: ¿Quién lo desarrolló? ¿Cómo y cuántos son? ¿Usan su tiempo libre para proporcionarnos estas aplicaciones? Podemos enterarnos de algo mirando los créditos, pero me gustaría exponer una pregunta: ¿Si conocieras a los programadores de tu aplicación favorita, no les invitarías una taza de café? Yo sé que esto quizá es improbable, pero si donamos unas monedas a los desarrolladores del software libre tal vez ellos en algún momento puedan disfrutar esa taza de café o un buen trago a nuestra salud... y finalmente quien se beneficia es la comunidad.
¿Ustedes que opinan?

Mi experiencia en Linux

Cristiam Edwin Barreto

Desde hace algún tiempo en la empresa en donde trabajaba (y hablo de 10 años atrás) se usaba Linux, una ingeniera me lo enseñó una vez, me pareció interesante y vi un par de funcionalidades y un programa que utilizaba la empresa, que en ese entonces era a base de Linux; y hasta ahí llegó el tema, puesto que por más que quisiera estudiarlo a fondo no tenía un PC en ese momento.

Fue solo hace un tiempo que un amigo me comento del Linux Ubuntu y su escritorio en 3D, y que además de todo era un software libre. En ese momento fue cuando decidí incursionar en Linux, lo primero que hice fue instalarlo desde Windows, para aprender a manejarlo, no fue muy difícil acostumbrarme a los comandos, puesto que mi inicio con los computares fue a través de D.O.S. y es algo similar, cambian los comandos pero el fin es similar.

Inicialmente, tuve inconvenientes con la configuración de mi tarjeta gráfica, y tuve que reinstalar Ubuntu repetidas veces, puesto que cada vez que instalaba los controladores de mi tarjeta gráfica lo desconfiguraba totalmente.

Pero después de estudiar y consultar en Internet, y ver la gran variedad de soporte que hay para Linux Ubuntu, solucioné el inconveniente y pude al fin configurar de manera correcta la aceleración 3D y también los juegos que tiene Linux en 3D, adicionalmente también algunos juegos que son de Windows los configuré correctamente, las demás partes de mi PC, se configuraron automáticamente, se puede decir que lo únicos inconvenientes que tuve fueron con mi tarjeta gráfica y de TV.

Un gran inconveniente que tuve luego de instalar Ubuntu, es que utilizo una herramienta de Windows, la cual es el Internet Explorer, Ubuntu tiene Mozilla, pero hay páginas que utilizo para estudiar que solo funcionan con Internet Explorer.

Y para solucionar este inconveniente investigué a través de la red, y descubrí una excelente aplicación para Linux, "Virtual Box" a través de la cual pude instalar Windows dentro de Ubuntu y utilizar no solo la aplicación que necesitaba, sino todas las demás que trae Windows, y así dejar de depender de este.

Según veo Linux solo tiene un inconveniente y es que mucho hardware viene con controladores únicamente para Windows, y para poder configurar adecuadamente algún hardware en Linux hay que investigar mucho. Pero todo esto lo contrarresta el gran soporte que recibimos en la red para usuarios Linux.

Para reflexionar:

"No paguemos por algo que debe ser gratuito, paguemos dando soporte e invitando a que más y más personas usen Linux."

¿Te interesa colaborar con Papirux?

Puedes enviarnos tus ideas, propuestas, artículos, opiniones a nuestra dirección de correo:

papirux@papirux.org

También puedes contactar con nosotros mediante **IRC** en el canal **#papirux** en **freenode.org**, o suscribirte a nuestro grupo de correo en **GoogleGroups**.