

# Acercamiento al Usuario Innovaciones recientes en Linux

28 de septiembre de 2002

Esta charla, por impartir en la Vigésima Primera Convención de Estudiantes de Centroamérica y Panamá en Montelimar/Nicaragua, tiene un título engañoso, y un subtítulo . . . engañoso: El contenido no es dirigido a usuarias, sino a técnicos, dando a conocer razonamientos y decisiones de diseño detrás de ciertos comportamientos “diferentes” en comparación con un sistema muy conocido. Las innovaciones presentadas en parte no son recientes y en su totalidad no restringido a Linux.

Sin embargo es reciente, que la combinación del avance tecnológico en hardware, impulsado entre otros por la industria tecnológica del usuario final en combinación con políticas de comercialización de la empresa dominante de Software de usuario final, y el evolvimiento imparable de soluciones del Software Libre en aplicaciones y sistemas maduros esta por marcar obsoleto la dependencia de la computación personal de soluciones propietarios.

Ahora la charla está disponible en red en [http://www.magma.com.ni/~jorge/linux\\_inova.html](http://www.magma.com.ni/~jorge/linux_inova.html), donde también se encuentran los diapositivos de la presentación.

El documento está también disponible en format PDF, al igual los diapositivos.

## Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Casos de uso comunes en la PC</b>	<b>:40 3</b>
2.1. Autenticación – ¿quién se llama raíz? . . . . .	3
2.2. Medios extraíbles – el disco agitado . . . . .	4
2.2.1. Unidades que nos juegan letras . . . . .	4
2.2.2. Llaves crecent para computadoras . . . . .	6
2.2.3. Mecánicos automatizados . . . . .	7
2.3. Chequeo del sistema de archivos – preludeo . . . . .	7

<b>3. Journaled Filesystems</b>	<b>:20</b>	<b>8</b>
3.1. Escenarios de desastres – ¡se congeló!		9
3.2. Prevención del caso peor – toma nota		9
<b>4. Aplicaciones Punta</b>	<b>:25</b>	<b>10</b>
4.1. Correo – la evolución no se puede detener		10
4.2. Navegación – Dinosaurios?!		11
4.3. Oficina – ¿a donde quieres llegar hoy?		11
<b>5. Instalación</b>	<b>:10</b>	<b>12</b>
<b>6. Transmigración</b>	<b>:10</b>	<b>13</b>
<b>7. Notas de Edición</b>		<b>14</b>

## 1. Introducción

El evolvimiento de las redes globales está disminuyendo la brecha entre las usuarias finales, que en su tiempo de surgimiento usaron computadoras de juguetes y para jugar, o terminales de caracteres en sistemas timesharing gigantes, y las administradoras de sistemas, que conocen cada script de un sistema tan íntimamente como el mecánico automotriz la gota de aceite en el tornillo que sujeta el bushing del cigüeñal.

Hasta los sistemas operativos de Microsoft le están exigiendo tareas de administración del sistema a la usuaria e incluyen desde el inicio conceptos de co-computación en redes con diferenciación de usuarias. Por otro lado, y esto se deben en su mayor parte al regalo que hizo Linus Torvalds al mundo informático al darle Linux, la aplicación en masa de conceptos de computación Unix hizo surgir la demanda por simplificación del manejo y de la administración de estos sistemas. Mientras una instalación de un sistema comercial de Unix hace quince años duro un día entero y solo pudo ser efectuado por una especialista, hoy en día una usuaria competente puede escoger entre varias variantes de Linux que instalan en dos a cuatro horas, sin ni siquiera requerir un manual impreso.

En la charla nos enfocamos estrictamente al ámbito de la *computación personal*, o computación de *escritorio*, quiere decir, el escenario de una computadora de “mediano” poder de cómputo y “bajo” precio, que es usada por una sola persona para efectuar trabajos mediante aplicaciones de alto nivel de abstracción del sistema y que no provee servicios de redes o lo hace en menor escala.

Este ámbito está marcado por el sistema “Windows” de la empresa Microsoft en sus múltiples versiones y variantes y comúnmente se supone, que sistemas alternativos tienen que presentar las mismas características para competir el segmento de mercado.

Esta charla trata de seguir algunas pistas, del porqué Linux es diferente, porqué seguirá siendo diferente en algunos casos, y porque ya no es diferente en otros.

No queremos dejar de mencionar qué también tradicionalmente existe un segundo segmento en el mercado, ocupado por los sistemas MacIntosh, empresa que recientemente está innovando en dos sentidos: Abandono completo de su sistema operativo a favor de un sistema basado en tecnología en Unix – Steve Jobs ha sido siempre un iluminado, posiblemente hay quienes recuerdan la chispa de corta vida NeXT-Step – y está abandonado la obsoleta categoría de ingeniería de sistemas propietarios, abriendo el código fuente de su sistema.

## 2. Casos de uso comunes en la PC

:40

### 2.1. Autenticación – ¿quién se llama raíz?

Usuarios de MS-Dos o Windows estaban acostumbrados a encender la computadora y comenzar a trabajar. Hasta el momento que los sistemas se integraban a redes había que utilizar una autenticación como usuario, dando un nombre y una clave al diálogo, antes de poder acceder al “Escritorio”. En las primeras versiones de la red Microsoft, que todavía tiran sombras hasta los días actuales, no hay *administración centralizada* de usuarios. En el momento que cualquiera ingresa unos caracteres en la casilla, y si esta cadena no está conocida al sistema, se crea un nuevo usuario con el nombre introducido.

Esta conveniencia no se puede mantener cuando se comparten recursos entre diferentes actores y cuando estos actores no se mantienen siempre en *el mismo lugar*. Para asignar un recurso específico a una identidad específica se requiere determinar la identidad de una usuaria en el momento que accede a “la red”, y la famosa ventana de login se hace imprescindible.

En concepto de detalle: la administración centralizada, introducida por Windows NT, permite a varias usuarias tomar control de la asignación de (derechos de uso de) recursos a identidades de usuarias, en Unix este derecho está concedido a una identidad resaltada, llamada tradicionalmente “root” o raíz. Este es el famoso Superuser. ¿Porqué solo uno? Keep It Simple and Stupid – KISS.

Sistemas Unix arman el mecanismo de autenticación mediante programas dedicadas *después* del inicio del sistema. Por lo tanto no es obligatorio hacerlo, sino simplemente “good practice” - práctica razonable. Ningún vendedor de un sistema Unix, aunque se llame Linux puede permitirse omitir esta práctica para facilitar el acceso a la computadora, porque perdería instantáneamente cualquier buena reputación. Por mi parte lo he hecho y puede ser encontrado en <http://www.magma.com.ni/~jorge/lios/>, solo por curiosidad.

Por otro lado, se han implementado medidas, que permiten de manera más controlada el acceso a una computadora, sin necesidad de autenticación explícita. El gestor de display – este es el mecanismo que realiza la autenticación de usuaria en ambiente X-Windows – que se provee para el Gnome Desktop: gdm, puede ser configurado de tal manera, que realiza un login automático para una usuaria del sistema específica instantáneamente cuando arranca el sistema, o después de un tiempo de espera. Esto puede ser usado en casa,

donde solo hay una usuaria en el equipo, o en computadoras compartidas entre personas anónimas (Internet Café, p.ej.), donde se crea una usuaria “visita”, con privilegios de acceso al sistema bien seleccionados: Cualquiera puede usar la cuenta “visita”.

La desventaja se presenta, cuando una usuaria diferente a la usuaria estándar quiere usar la máquina, tiene que cerrar primero la sesión, y después abrir una nueva con la otra identidad.

## 2.2. Medios extraíbles – el disco aguado

### 2.2.1. Unidades que nos juegan letras

Los sistemas Microsoft heredaron desde CP/M de Digital Research el concepto, que la usuaria tiene que administrar los dispositivos de almacenamiento de datos: Discos, Floppies, CDRoms, etc. Cada dispositivo recibe una letra mayúscula como identificador, y para referirnos a un objeto de archivo en un determinado medio utilizamos la notación:

`LetraUnidad:ViaDeAcceso`

p.ej:

`A:\conescapan\charla.doc`

Este sistema funciona muy bien, mientras la cantidad de dispositivos no excede ciertos límites, para hablar precisamente: 26 unidades, caso que es válido en casa, pero no en una red, donde recursos compartidos pueden exceder sin grandes problemas este límite.

La reducción del espacio de nombres con una sola letra crea el problema que la usuaria tiene que recordar o anotar, qué significado tiene cada letra. Por suerte la Red Microsoft introdujo otra nomenclatura:

`//Servidor/RecursoCompartido/ViaDeAcceso`

que reduce significativamente el problema mnemónico.

Donde quiero enfocar este capítulo sin embargo es otro problema, los medios extraíbles. En este caso tenemos una unidad (A:), que maneja diferentes *medios* en diferentes momentos. Cada medio contiene un *sistema de archivos* con su respectiva información individual – al cambiar el medio, cambia la información “en” la unidad.

Mientras los sistemas operativos fueron pequeños las veces que se accedía a un objeto (fichero) en el medio, el sistema volvía iterar sobre la completa vía de acceso en el sistema de archivos para obtener la parte solicitada de su contenido. Esto requiere varios accesos

a la unidad, que en el caso de unidades con medios baratos hoy en día resulta en retrasos sumamente ridículos.

Un floppy disk lleva un volumen de 1.44MByte de datos. En una computadora pequeña, casi obsoleta con 64Mbyte Ram, pueden almacenarse 44 Floppy disk en Ram. Un Flash-Card con 16Mbyte cuesta lo mismo que la unidad Floppy y podría almacenar el contenido de diez Disquetes. Por otro lado, el acceso a estos 1.44 MByte de datos en un Floppy Disk se ejecuta con los mismos 400-800kByte/s como en los tiempos viejos. Mientras escribir este volumen de datos en un disco duro tarda unos milisegundos, necesitamos más que un minuto para grabarlo en un Floppy.

La técnica de usar caché – memoria intermediaria – para aumentar el acceso ha sido introducido en todos los sistemas operativos desde siempre, que hubo suficiente memoria para no impedirlo. La idea es, cuando se accede a alguna parte del contenido del floppy guardarlo en la memoria, con la suposición que la aplicación (la usuaria) volverá muy probablemente acceder y/o manipular a esta misma información. Al determinar que el acceso a este contenido ha terminado, se guardan los cambios realizados de nuevo en el disco. Esto no solo se aplica a contenido de archivos, sino también a *meta-datos* del sistema de archivos: atributos y nombres de archivos, tablas de bloques libres, arboles de directorios, etc.

El cuello de botella – escritura/lectura secuencial a un medio lento – se puede reducir significativamente con este método, pero en el caso de los floppy disks la conveniencia tiene que ser comprada con el precio de la posible corrupción de datos:

Para mayor simplicidad en la producción de las unidades, la introducción y expulsión del medio (floppy disk) es realizado por la usuaria. Para evitar la extracción durante el mismo proceso de lectura o escritura se cuenta con una lámpara, que indica el “uso” del medio, y se indica a la usuaria, que no hay que extraer el disco, mientras la lámpara está encendida. Sin embargo, con el caché de datos en memoria, puede no estar en uso la unidad, pero los datos en el floppy no coinciden con el supuesto contenido – ya que este fue modificado en memoria.

Consideramos el escenario de Jorge, que busca el documento de la charla para Conescapan en un disquete, lo inserta, abre un archivo y se da cuenta, que es una versión obsoleta. Ahora retira el disquete, sin cerrar el archivo, inserta otro disquete, abre un archivo con el mismo nombre y se da cuenta que es la versión que busca.

Por error, utiliza la ventana equivocada al agregar unas líneas y cambia el archivo “viejo”. Se da cuenta y cierra la ventana correspondiente, momento en el cual el sistema intenta guardar los cambios realizados en el archivo en el disquete – que contiene la versión nueva de la charla.

En el peor caso, el sistema operativo sobrescribe el archivo “bueno”, con la versión obsoleta modificada, pero por suerte, este caso hoy en día ya no se da. En caso de los sistemas Microsoft es común, que en este momento la pantalla se convierte en el color azul conocido por todos los usuarios y muestra un mensaje de error indicando que el

disco insertado tiene un número serie incorrecto y que hay que insertar otro disquete con el número correcto.

Apple Computers ha reconocido la particularidad de este caso de uso ya muy temprano, y sus computadoras “tragan” los floppies, y los devuelvan al usuario mediante un mecanismo controlado por la computadora. Por supuesto, antes de expulsar el floppy para dar lugar a otro, actualizan el primer floppy automáticamente.

### 2.2.2. Llaves crecent para computadoras

Los sistemas Unix introducen un sistema de archivos unificado: al arranque del sistema se utiliza un medio predefinido – el *root-filesystem* – que contiene las utilidades de sistemas. La raíz del directorio de este medio, forma la raíz del sistema de archivos unificado.

Si se quiere acceder a otro medio: disco duro, floppy, cdrom, ramdisk, etc. se utiliza un subdirectorío (opcionalmente vacío) como *punto de montaje* el cual se asocia a través del registro de la unidad correspondiente en el sistema con la raíz del sistema de archivos en el medio respectivo. Este proceso se llama *montar* una sistema de archivo. A partir del montaje todo el contenido de un medio está disponible como árbol de directorios contenido en el subdirectorío que sirve de punto de montaje. Este mecanismo se usa uniformemente entre unidades fijas, extraíbles, y recursos compartidos en red.

Consecuencias: No hay letras de unidades, el espacio de nombres es libremente elegible por la usuaria, todas las utilidades trabajan uniformemente en todas las unidades montadas.

Para acceder a un medio extraíble se inserta este en la unidad correspondiente, y se monta en un determinado punto de montaje. A continuación se accede al subdirectorío correspondiente para acceder y modificar el contenido del medio.

Antes de extraer el medio, se tiene que invertir la operación – se *desmonta* la unidad. El montaje/desmontaje forma puntos definidos, donde el sistema operativo obtiene metadatos (el directorío) sobre el medio en cuestión, los pone en caché, o actualiza en el segundo caso los datos guardados en el medio con sus modificaciones en memoria.

Para usuarias de Windows este mecanismo tiene dos consecuencias irritantes:

- Hay que ejecutar un comando antes de acceder a un medio extraíble (adicionalmente a la tarea difícil de insertarlo en la unidad)
- Aunque se extrae el medio de la unidad, la computadora todavía piensa que el medio está insertado: Hay que ejecutar un comando antes de extraer un medio de su unidad.

Efectivamente, esto es una de las causas mayores de confusión y errores en el uso de computadoras Linux, aunque el método Unix es en principio más seguro contra pérdida de datos que el método anterior descrito, ya que al insertar otro medio sin desmontar el primero, el sistema se rehúsa a accederlo.

### 2.2.3. Mecánicos automatizados

Varios enfoques han sido desarrollados para simplificar esta situación del punto de vista del usuario, lastimosamente: Hay una diversidad de propuestas, tanto para la nomenclatura del punto de montaje, como para los mecanismos para (semi-)automatizar el proceso de montaje/desmontaje de medios extraíbles, que al final la situación es más confusa todavía. A esto se suma, que las unidades por tradición y consideraciones de seguridad solo pueden ser accedidos desde el superuser, situación que tiene que ser circunvada en el caso de computadoras personales y lo que puede ser realizado de diversas maneras.

La única solución efectiva es la extinción de los floppy disks, y esperemos que sea pronto, o su modificación para incluir un mecanismo confiable de detección de cambio de medio y de bloqueo de entrega.

He aquí algunos de las soluciones “mientras tanto”:

**Montaje síncrono** este término significa, que el caché de escritura es desactivado en la unidad respectiva: cuando una aplicación modifica el contenido de un disquete, estos cambios son escritos instantáneamente en la unidad, mientras la lectura se efectúa usando las ventajas del *caching*. De esta manera, no puede haber destrucción por inconsistencia de contenido.

**Auto-desmontaje** Esto funciona en combinación con el primer mecanismo. Cuando al acceder al medio se detecta que el disco ha cambiado – algunas unidades pueden señalar eso – se desmonta el floppy anterior. Ya que está sincronizado, no hay caché pendiente de escritura.

**Auto-montaje** Este mecanismo fue desarrollado para redes grandes y complejas con muchos recursos compartidos. Un servicio del sistema (daemon), que maneja un registro de unidades con sus puntos de montaje correspondientes vigila los accesos a estos últimos. En caso que una aplicación quiere acceder al contenido de un punto de montaje, el daemon efectúa primero la operación de montaje respectiva. La aplicación se bloquea mientras la unidad está lista. Si la usuaria no ha insertado el medio, se efectúa un error de acceso.

**Floppy-daemon** Un servicio de sistema accede periódicamente a la unidad (p.ej. cada segundo), para comprobar si hay un medio insertado. En caso positivo se monta el medio. En caso de haber extraído un medio antes montado se desmonta el punto de montaje.

## 2.3. Chequeo del sistema de archivos – preludeo

Todos los sistemas operativos modernos utilizan disk-caching extensivamente, así como la extensión de la memoria física con un espacio de intercambio (Swap Space). Esto

conlleva, que constantemente muchos datos, que son parte del contenido de archivos o meta-datos se encuentran en memoria durante la operación normal del sistema. Ya que los días gloriosos en los cuales la memoria principal consistía de núcleo de ferrito y mantenían su contenido al apagar la computadora son pasado, se introdujo la operación “shut-down” del sistema, que consiste en sincronizar todos los caches con el disco duro. La interrupción de energía imprudente puede tener consecuencias catastróficas en los datos y en el sistema de archivos de un disco duro, hasta imposibilitar su uso.

Mientras en computadoras con Windows cada usuario puede apagar la computadora, esto puede tener consecuencias fatales para usuarios en red, que utilizan recursos compartidos desde esta máquina. En sistemas Unix tradicionalmente la operación de apagar el sistema es reservado para el superuser, por razones KISS y consideraciones parecidas. Los sistemas Unix no solo son multitarea, sino también multiusuario y permiten sesiones de varias usuarias simultáneamente. ¿Cuál de ellas tiene el derecho de apagarles a las otras la computadora?

En nuestro caso de las computadoras personales esto es una pregunta no aplicable, solo hay una usuaria, que no quiere molestar en acceder como superuser al sistema.

En modo texto/consola, al menos existe tradicionalmente el método Ctrl-Alt-Supr para reinicializar la computadora, aunque esto significa, que la usuaria tiene que esperar el proceso y apretar el botón en el momento oportuno, antes que la computadora reinicializa el sistema.

En modo gráfico, otra vez nos ayudan los administradores de Display, que hoy en día proveen un menú de sistema, con las opciones de re-inicializar y apagar el sistema.

Para personas más impacientes existen aplicaciones de menú con la misma función (como en Windows), pero que posiblemente piden por lo menos a la usuaria la clave superuser y proveen de esta manera una pause de re-capacitación adicional. El permiso de apagar el sistema sin pedir clave puede ser accedido sin embargo también con utilidades como “sudo”, “calife”, o “super”.

Otra diferencia en el uso de la computadora en casa es, que los sistemas de archivos Unix esfuerzan el chequeo del sistema de archivo periódicamente, por lo general el mismo día que la persona respectiva no tiene los quince minutos disponibles, que requiere este proceso. A pesar de cerrar bien el sistema, las aplicaciones, fallas en el disco mismo, en la memoria o en rutinas del sistema operativos pueden causar incoherencia en los datos o meta-datos guardados en el disco duro. El chequeo periódico es un mecanismo para la detección y corrección de estas fallas.

### **3. Journalled Filesystems**

:20

Este capítulo continúa la noción del anterior en cuanto al chequeo de sistemas de archivos. Journalled Filesystems eliminan el requerimiento del chequeo completo del disco duro en caso de un cierre abrupto del sistema en la sesión anterior.



Linux tiene soporte (por lo menos) para:

**Ext3** una extensión del sistema de archivos “estándar” de Linux Ext2. Es un sistema de archivos Unix tradicional con muchas mejoras en cuanto a velocidad de acceso y seguridad.

**XFS** es el sistema de archivos estándar para los sistemas Irix de la empresa SGI.

**ReiserFS** es un sistema de archivos revolucionario, que deja de contemplar el disco duro como una secuencia de bloques, y utilizar B\*Trees para los meta-datos. ReiserFS provee un acceso más rápido al contenido, y liberar alrededor de 8 % más del espacio del disco duro en comparación con los sistemas de archivos tradicionales.

### 3.1. Escenarios de desastres – ¡se congeló!

En caso de que un chequeo del disco al iniciar – sea provocado o programado – detecta fallas en el sistema de archivos, se intenta realizar una corrección, sin embargo muchas veces los cambios a realizar no pueden realizarse automáticamente, especialmente si no hay información redundante para realizar la reparación, o si la reparación involucra cambiar el contenido de un archivo o directorio. Supongámonos la situación, cuando la usuaria está borrando un archivo, y en este momento falla la energía. Posiblemente ya están registrados los bloques en la lista de bloques libres, pero todavía existe una entrada en el directorio. El sistema no se atreve a decidir por propia voluntad, si el archivo ha sido creado y los bloques libres todavía no se han de-registrados, o si la situación fue la inversa.

En este momento se interrumpe el chequeo del disco, y se le pide a la usuaria efectuar un chequeo manual, de la partición en cuestión. Esta operación requiere autorización de superuser, y obviamente también sus conocimientos. . .

### 3.2. Prevención del caso peor – toma nota

Un Journalled Filesystem convierte los accesos al disco duro en “transacciones”, un concepto heredado de los bases de datos. En una especie de “block de notas” – el Journal – se registran primero las modificaciones a realizar en el sistema de archivo, después se ejecutan punto por punto, y al terminar de ejecutar un paso, se marca terminado en el journal.

De esta manera, se puede garantizar un estado consistente – o *ante Journal* – o *después Journal*. En caso de una falla de energía, durante el proceso de arranque se ejecuta los pasos no terminados en el Journal, o se revocan los pasos ya efectuados. para llegar a uno u otro de los dos estados.

El tamaño del Journal determina la cantidad de transacciones que pueden estar pendientes, y por lo tanto la velocidad de acceso para cargas altas del sistema. Más grande el

Journal, más rápido el acceso al disco duro. Por otro lado, en caso de falla, el tiempo del *replay* – la recuperación de un estado consistente mediante el Journal – es proporcional al tamaño del Journal.

Los discos de hoy – 120 GByte y creciendo – toman hasta 20 minutos para un chequeo completo. Un Journal de 20 o MByte provee un buen rendimiento y solo toma algunos segundos para el *replay*.

Así no solamente se reduce la intervención humana en la computadora, sino también se mejora la disponibilidad o reduce el “down-time” respectivamente.

## 4. Aplicaciones Punta

:25

El mito actual de Linux es, que es muy difícil de manejar, y que las aplicaciones Windows son mucho más sencillas. Esta aserción ya no tiene razón, en cuanto se refiere a aplicaciones comunes. Algunos de ellos se presentan en este capítulo, pero primero se presentan algunos datos que para toda eternidad van a hacer “más difícil” a Linux que a sistemas propietarios:

**Hay mas que una manera** El desarrollo de Software Libre no es controlado por una fuerza coercitiva, que determina las funciones disponibles. En cuanto a ambientes gráficos tenemos por lo menos dos alternativas: Gnome y KDE, que están a la punta en tecnología de interfaces gráficos de usuarios. Una de sus características es, que pueden cambiar completamente su apariencia, sin cambiar su funcionalidad. . .

Esto obviamente hace más difícil el entrenamiento, y eleva el nivel de entrada para usuarias novatas: No hay *un* Linux.

**El ambiente gráfico es solo una aplicación** Mientras Windows y MacOS integran completamente el ambiente gráfico en el sistema operativo, Linux (Unix) es completamente inerte acerca la escogencia del interfaz con el mundo. Entre miles de usos para Linux – desde controladores industriales, hasta motor para computadoras de juegos, hay uno, que es él de la computadora personal. Por lo tanto existe una variedad de aplicaciones, que no tienen interfaz gráfico, porque no es apropiado.

**Mayor exigencia técnica** Linux viene del ambiente de servidores net multiusuario y siempre va a contribuir como factor importante en ellos. Esto requiere que tenga características correspondientes, que exigen mayor conocimiento para su configuración y manejo que un sistema carente de estas características.

### 4.1. Correo – la evolución no se puede detener

Lo que es Outlook (Express) en el mundo Microsoft, tiene ahora su colega en el mundo del Software Libre. “Evolution” es un agente de mensajería integrado con funcionalidad muy

parecida a este programa. La empresa que contribuye en primer lugar en su desarrollo: Ximian, también ofrece un producto, que permite que Evolution interactúe con servidores de correo Exchange de Microsoft.

Recientemente Microsoft anunció un cambio en sus políticas de Licencias, y Evolution es manejado ahora como una alternativa oportuna para la sustitución de productos correspondientes en ambientes comerciales con servidores basados en Windows NT.

Evolution forma parte del ambiente Gnome Desktop, pero en especial, Ximian mantiene una versión manejada de Gnome, llamado Ximian Desktop, con muchas mejoras en cuanto al interfaz gráfico. Naturalmente Ximian Desktop y Evolution pueden ser obtenidos sin pago de Licencias desde la red.

## 4.2. Navegación – Dinosaurios?!

La empresa Netscape, que produce el navegador HTML más popular, o segundo más popular en le mundo hace años ha emprendido el camino, de abrir el código fuente de su producto.

Hace poco esto culminó – pero no terminó – con la versión 1.00 del navegador Mozilla, que corresponde a la versión 6 del producto comercial.

El objetivo del proyecto Mozilla es, proveer un Navegador multiplataforma, empresa muy costosa si es realizado bajo el paradigma del software propietario.

Un producto de la apertura de Netscape son navegadores livianos como Skipstone y Galeon, que han sido desarrollado re-utilizando componentes de Mozilla, pero no presentan la demanda de recursos de este.

## 4.3. Oficina – ¿a donde quieres llegar hoy?

Un camino parecido ha tomado el producto StarOffice, que es un paquete Office inclinado hacia la compatibilidad con Microsoft Office. La empresa alemana StarDivision fue comprado en su totalidad por Sun Microsystems, donde poco después se tomó la decisión de abrir el código fuente del producto.

En Julio de este año se publico la versión 1.0 de OpenOffice, un paquete moderno y popular que está disponible en 18 idiomas en las plataformas Sun-Solaris, Linux-Intel, MacIntosh y Microsoft-Windows.

OpenOffice puede usarse como sustituto inmediato del suite de Microsoft con editores compatibles para:

1. Documentos Word
2. Hojas de Cálculos Excel

### 3. Presentaciones Powerpoint

y que además provee editores de HTML, de dibujos y de imágenes, así como un navegador integrado, un ambiente de mensajería parecido a Outlook con agenda que puede ser usado en red.

## 5. Instalación

:10

Las formas de instalación de un sistema Linux son incontables. Distribuidores como RedHat y Mandrake realizan una selección cuidadosa del grupo meta y de los equipos y accesorios soportados de tal manera, que la mayoría de las usuarias pueden realizar una instalación con mínima intervención, ya que los gestores de dispositivos pueden autodetectar la configuración del sistema, y la selección del Software y su configuración son predeterminadas. La desventaja de éste método es, que equipos que no concuerdan con las características predeterminadas de la distribución respectiva no tienen soporte directo y no pueden ser instalados, o solamente con el esfuerzo usual en otras distribuciones. También usos que no concuerdan con los previstos por el distribuidor son algo más difíciles de configurar.

Cabe señalar aquí, que una gran diferencia entre sistemas operativos propietarios y sistemas GNU/Linux es, que, al menos en teoría, cualquier aplicación puede ser instalado y configurado en cualquier distribución, por ende consiguiendo el código fuente y recompilándola.

Las distribuciones como Slackware y Debian han tenido un enfoque diferente, que por un lado favorece a la instaladora interesada en determinar cada detalle de la configuración de antemano, y permite así sacar el provecho máximo de las facilidades del sistema – hardware y aplicaciones – pero por otro lado requiere un vasto conocimiento sobre computación, sistemas operativos y el hardware usado.

Para instalaciones a nivel institucional existen herramientas poderosas, como por ejemplo FAI - Fully automated Installation, que es inspirado por un producto de Sun-Microsystems, y permite a un administrador de sistemas pre-configurar todo el Software y las configuraciones individuales de cuantas computadoras desee en un servidor de instalación. El único requisito para la instalación o re-instalación de una computadora es un Boot-Prom, que puede ser programado con el mismo Software, o en falta de esto un floppy disk de arranque, que activa la tarjeta de red y comienza a bajar del servidor de instalación todo software necesaria para instalación y configuración. En computadoras actuales con recursos decentemente suficientes un proceso de instalación puede durar entre media hora y una hora.

Hay que destacar aquí también, que la instalación de un sistema Linux no es comparable con la de un sistema operativo de Microsoft Windows: el último instala los servicios básicos del sistemas y algunas aplicaciones y tiene en la actualidad una duración aproximadamente de 45 minutos a una hora y media. Después hay que instalar y configurar las aplicaciones oficina y otras.

Las distribuciones Linux no solamente incluyen el sistema operativo, sino también todas las aplicaciones disponibles en la distribución, en caso de RedHat alrededor de mil paquetes de Software y en caso de Debian ascendiendo a casi nueve mil. La instalación, con una duración parecida o hasta el triple del tiempo instala y configura también una cantidad arbitraria de estas aplicaciones.

Alrededor del último “LinuxTag” en Alemania salió el CD “Knoppix”, que tiene un enfoque levemente distinto. Es un CD arrancable con sistema de archivos comprimido, así que alberga alrededor de dos Gigabyte de aplicaciones en un CD de 650 Mbyte. Al iniciar el CD carga un sistema Linux que auto-detecta el hardware disponible y en vez de instalar un sistema Linux lo ejecuta directamente desde el CD. Knoppix puede ser utilizado en una gran variedad de computadoras PC, sin tener efecto alguno sobre el sistema ya instalado, así que puede uno usarlo en las computadoras de bibliotecas, donde amigos que todavía no usan Linux, etc.

DemoLinux es otro sistema con características parecidas, que es desarrollado por un grupo de programadores en Francia.

La empresa antecesor de Ximian – Progeny, creó un instalador gráfico, basado en el sistema de empaquetamiento de Debian y Ximian mismo vertió la experiencia en Red-Carpet – la alfombra roja, un sistema de actualización de instalación gráfico, que avisa a la usuaria conectada a la red de actualizaciones disponibles de su sistema y los puede realizar en línea y sin interrumpir las tareas en curso.

## 6. Transmigración

:10

Hace pocos años el argumento más escuchado en cuanto a la comparación entre los sistemas propietarios y Linux ha sido: “Windows tiene programa X. ¿Linux también lo tiene?”.

Esta situación está por cambiar en el último tiempo. El paquete OpenOffice es un ejemplo, donde un producto semi-propietario ha sido usado como base para crear un Software Libre. En la actualidad el desarrollo se está “invirtiendo” y las mejoras del producto se vierten en el producto comercial StarOffice que todavía existe y con la buena razón de darle un producto con soporte profesional y garantía a empresas e instituciones que así lo requieren. Un proceso parecido ha tenido el navegador famoso “Netscape”, cuyo nombre de proyecto “Mozilla” es el nombre del mismo producto en su forma de código abierto. Mozilla es basado en el código fuente original de Netscape, pero con un enfoque de compatibilidad en todas las plataformas que disponen de un compilador C y un ambiente gráfico. Una inversión comercial en una extensión de un programa de este estío tuviera costos prohibitivos.

En el campo de los juegos por computadora vale mencionar dos bibliotecas de programación: Allegro y SDL. Allegro provee al programador un conjunto de funciones gráficos, de sonido, y de manipulación de Joysticks, etc., en corto, todo lo que se necesita para la

## 7 Notas de Edición

programación de juegos gráficos en tiempo real. La biblioteca está disponible para MS-Dos, Microsoft-Windows, el ambiente de consola virtual de Linux, el ambiente gráfico de X-Windows, y por lo tanto un juego programado con Allegro puede ser compilado y ejecutado en cualquiera de estos ambientes.

SDL – Simple DirectMedia Layer provee una funcionalidad parecida y se usa para los mismos fines.

También la rutina de interfaz gráfico Gtk – Gimp ToolKit, que se está usando en el ambiente de escritorio gráfico Gnome está disponible en una versión para Windows, así que existen ya varias implementaciones de Software, que han originado en el ambiente de Linux y ahora están también disponible en Windows.

Una dificultad para la compilación de aplicaciones realizadas bajo un sistema Unix bajo Microsoft Windows, es la falta de rutinas y llamadas al sistema compatibles. Cygwin es un ambiente de programación completo bajo Windows, que emula todos los componentes necesarios para que exista solo un esfuerzo mínimo en la adaptación de los códigos fuentes en C o C++ para su apropiada ejecución y compilación bajo Microsoft Windows. Con una instalación de cinco minutos los amantes a los comandos shell del interpretador bash pueden obtener su ilusión del mundo perfecto en una ventana.

## 7. Notas de Edición

Para la edición de los documentos estoy usando el editor Lyx que tiene excelente respaldo y documentación en español y para el cual existe una versión en Windows también.

Utilizo la clase FoilTex para los diapositivas, en formato  $\text{\LaTeX}$ , documentos generales escribo generalmente con la clase KomaScript-Article. Tengo intenciones de utilizar la clase Prosper, para los diapositivas, pero todavía no tiene soporte en  $\text{\LaTeX}$ .

Magicpoint es otra herramienta muy útil, que tiene dos ventajas sobre lo anterior:

1. Mucho mejor representación y claridad de las letras en pantalla
2. Edición más simple y rápido de la presentación.

La desventaja es, que la conversión en formato escrito o en pdf es más difícil que en formatos  $\text{\LaTeX}$ .

Este documento Web se elaboró con  $\text{\LaTeX}$  y la clase Hyperlatex, utilizando la utilidad miserable Hyperlyx creado por mi para adaptarlo a  $\text{\LaTeX}$ .

El autor trabaja actualmente como asesor/director en el centro de operación de redes en la Universidad Nacional de Ingeniería en Managua/Nicaragua y en el Campus Virtual de la misma institución y como consultor y docente en informática y puede ser contactado en la siguiente dirección electrónica: Jorge.Lehner@gmx.net