

Free Tux

Por un Mundo Libre...



En este mes....

Un vistazo a Open SuSE 10.3

SUSE Linux

La última edición del camaleón (ahora más verde que nunca) ha sido de las más esperadas en la comunidad linuxera.

elitec.org



Novell



Y Hablaremos de:



Fedora



Xfce



Firewalls

y "Todo lo que un linuxero debe saber"

Gnu/linux Zone

Staff



Shadow

Estudiante en el área de Sistemas Informáticos.
Mexico
Editor



Menroot

Diseñador y Administrador de Sistemas.
Colombia
Escritor



Rodolfo

Ing. Telemático.
Rep. Dominicana
Corrector y Escritor



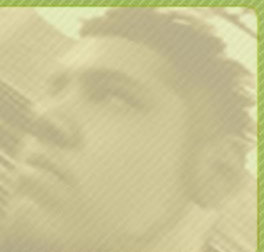
Lyonn

Estudiante en el área de Sistemas Informáticos.
Mexico
Corrector y Escritor



Zion

Estudiante en el área de Sistemas Informáticos.
Mexico
Escritor



GnDx

Seguridad Informática, Diseño y Programador web.
Eua - Mx
Diseñador

Editorial



Bienvenidos sean a esta cuarta edición de la revista Free Tux, su revista.

En esta ocasión Clitec agradece a la comunidad del software libre el gran apoyo que nos han mostrado descargando la revista y visitando el sitio.

Como sabrán algunos ya contamos con un sitio oficial <http://clitec.org> donde seguimos colocando ultimas noticias, además de nuevas secciones que con su ayuda, ya sea aportaciones y comentarios, irán creciendo poco a poco.

Este mes Free Tux trae una serie de artículos interesantes y de gran utilidad, como es la programación en Bash, la cual puede facilitarnos y ahorrarnos escribir todos esos comandos que usamos con frecuencia.

Aprenderemos algo de historia, conoceremos un poco más acerca de una distribución que en los últimos años se ha popularizado bastante que incluso se dice que puede llegar a ser más popular que Ubuntu, si yo se que si saben cual es, es nada mas y nada menos que Fedora.

Gn0x nuestro nuevo integrante, nos trae un artículo escrito por "zk" sobre algo que descuidamos mucho, la seguridad de nuestro equipo. Nos habla acerca de los firewalls algo que debemos de tener muy en cuenta para tener seguro nuestro sistema.

Maty un blogger que muchos han de conocer y también nuevo integrante, nos hizo favor de testear la distro del mes, OpenSuSe 10.3 en la cual nos detalla algunas de sus características y funcionalidad.

Además de nuestro sitio del mes, el cual se lo llevó por supuesto, Espacio Linux.

Queremos darle una cordial bienvenida a nuestros nuevos integrantes GndX y Osvaldo, a este equipo que siempre trata de aportar algo a los demás y que mejor haciéndolo con lo que nos gusta, el software libre.

Y también hacerles una invitación a todos los lectores a que se unan a este proyecto, como verán nos gusta trabajar y si nuestro trabaja ayuda a más personas, lo hacemos con gusto.



Indice

02 Staff

03 Editorial

04 Bash

17 Xfce

25 Fedora

30 OpenSUSE

38 EspacioLinux

39 Firewalls

43 Richard Stallman

50 Contacto

Bash

Sobreviviendo en GNU/Linux



```

bash-2.05b$ pwd
/home/dstone
bash-2.05b$ cd /usr/portage/app-shells/bash
bash-2.05b$ ls -al
total 68
drwxr-xr-x  3 root root  4096 May 14 12:05 .
drwxr-xr-x 26 root root  4096 May 17 02:36 ..
-rw-r--r--  1 root root 13710 May  3 22:35 ChangeLog
-rw-r--r--  1 root root  2924 May 14 12:05 Manifest
-rw-r--r--  1 root root  3720 May 14 12:05 bash-2.05b-r11.ebuild
-rw-r--r--  1 root root  3516 May  2 20:05 bash-2.05b-r9.ebuild
-rw-r--r--  1 root root  5083 May  3 22:35 bash-3.0-r11.ebuild
-rw-r--r--  1 root root  4038 May 14 12:05 bash-3.0-r7.ebuild
-rw-r--r--  1 root root  3931 May 14 12:05 bash-3.0-r8.ebuild
-rw-r--r--  1 root root  4267 Mar 29 21:11 bash-3.0-r9.ebuild
drwxr-xr-x  2 root root  4096 May  3 22:35 files
-rw-r--r--  1 root root   164 Dec 29 2003 metadata.xml
bash-2.05b$ cat metadata.xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE pkgmetadata SYSTEM "http://www.gentoo.org/dtd/metadata.dtd">
<pkgmetadata>
<herd>base-system</herd>
</pkgmetadata>
bash-2.05b$ sudo /etc/init.d/bluetooth status
Password:
* status:  stopped
bash-2.05b$ ping -q -c1 en.wikipedia.org
PING rr.chtpa.wikimedia.org (207.142.131.247) 56(84) bytes of data.

--- rr.chtpa.wikimedia.org ping statistics ---
 1 packets transmitted, 1 received, 0% packet loss, time 0ms
 rtt min/avg/max/ndev = 112.076/112.076/112.076/0.000 ms
bash-2.05b$ grep -i /dev/sda /etc/fstab | cut --fields=-3
/dev/sda1          /mnt/usbkey
/dev/sda2          /mnt/ipod
bash-2.05b$ date
Wed May 25 11:36:56 PDT 2005
bash-2.05b$ lsmod
Module              Size  Used by
joydev               8256  0
ipw2200              175112  0
ieee80211            44228  1 ipw2200
ieee80211_crypt      4872  2 ipw2200,ieee80211
e1000                84468  0
bash-2.05b$ █

p02p-S'02P# █
cT000                84400  0
!cccc00STT-cl0b#  4815  5 !bmSS00'!cccc00STT
!cccc00STT          44558  1 !bmSS00
!bmSS00              152115  0
!odqca              8520  0
!oq0jc              2156  026q p0
p02p-S'02P# 1200q
H0q 000 S2 TT:30:20 b01 S002
p02p-S'02P# q01c
x000x000x          3000x1000

```

Bash

Sobreviviendo en GNU/Linux



Primeramente tenemos que definir que es una shell. Una shell no es más que un interprete de comandos. En los sistemas operativos tipo Unix son muy usadas, ya que el verdadero poder está en la konsola. La shell funciona a grandes rasgos así: El usuario captura un "Comando" y la shell se encarga de revisar que éste exista, si existe lo ejecuta y envía una salida, en caso contrario informa acerca de un error.

Mencioné a un "comando", ¿Pero qué diantres es un comando? Un comando es una instrucción que mandamos al sistema operativo, el cual es sumamente claro y no permite ambigüedades.

Ahora que apenas vislumbramos el concepto de una shell pasemos a bash. Bash significa Bourne Again Shell. El padre es Stephen Bourne, pasando por Brian Fox y tiempo después por Chte Ramey, quien continuó el desarrollo de este ya viejo, pero muy funcional shell.

```

lyonn@localhost: /home/lyonn - Terminal - Konsole
Sesión Editar Vista Marcadores Preferencias Ayuda
[lyonn@localhost ~]$ ls
/          C++/      Documentos/  Fotos/     PERL/
amsn/     Descarga/  dollar/     GNUstep/  tmp/
amsn_received/ Desktop/   Firefox_wallpaper.png Musica/    Video/
[lyonn@localhost ~]$ whoami
lyonn
[lyonn@localhost ~]$
  
```

Un aspecto, que en lo personal se me hace muy interesante, es que está presente en varios sistemas operativos, tanto para Mac (no es raro, ya que es un sistema operativo tipo Unix) como para Windows con el proyecto Cygwin

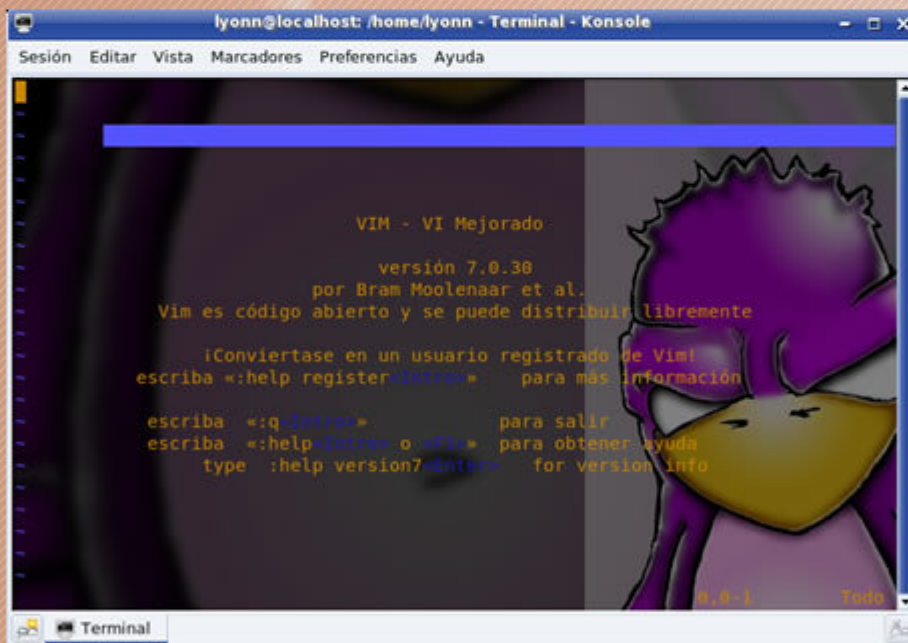
Bash

Sobreviviendo en GNU/Linux



A mero título de ejemplo, podemos nombrar otras shells como korn shell (ksh) y el C shell (csh), de las cuales bash tomó algunas ideas, como autocompletar con la tecla tab o el borrar historial (`$ cat >/bash_history`: Muestra el historial. Para borrar el historial es: `$ history -c`).
 Nota: `$` significa que el prompt está en modo usuario normal.

Bien, los scripts que presento no son grandes obras maestras, son cosas básicas que podemos automatizar y que nos quiten de estar haciéndolas comúnmente. El mes pasado se vió como sobrevivir en consola, así que no será un tema inexplorado y puede ser tomado como complemento.
 No queda otra que entrar en materia. Primero necesitamos un editor de textos, en lo personal me gusta emacs, vi, gedit, kate.



Pues hagamos nuestro primer ejemplo. Siempre tenemos que especificar `#!/bin/sh`, ya que éste indica que es un script de bash y que debe ser interpretado por él.

```

=====
#!/bin/sh
"Primer ejemplo, clásico Hola mundoiii
echo "¡Hola Mundo!"
=====
    
```

Bash

Sobreviviendo en GNU/Linux



Primero debemos guardarlo con la extensión `.sh`, yo lo nombré como `hola.sh`, lo siguiente que debemos hacer es otorgarle permisos de ejecución, eso lo hacemos con `chmod a+x hola.sh` y a continuación lo ejecutamos como `./hola.sh`

Mi salida:

```
>lyonn@localhost >>$ chmod a+x hola.sh
```

```
>lyonn@localhost >>$ ./hola.sh
```

```
"!Hola Mundo!"
```

Como observaran, la salida es `"!Hola Mundo!"`.

Ya pasamos el fabulofantástico ejemplo clásico de todos los lenguajes de programación, ahora veamos a las variables que tanto nos ayudan.

Variables.

Sin rodeos, las variables nos sirven para almacenar información en la memoria de la computadora, esta información pueden ser letras, números, direcciones de memoria, direcciones de archivos, nombres de archivos, etc.

Si queremos declarar una variable, por ejemplo, `km` con un valor de 100 sería algo así:

```
km=100
```

Observen los espacios (no hay), ya que `bash` es "sensible" a estos..XD, si lo pusieramos con espacios `bash` tomará cada cadena (`km`, el símbolo `=` y el valor) como un comando diferente.

Si quisieramos imprimir la variable `km` quedaría algo así:

```
echo $km
```

Si queremos que imprima un mensaje junto con la variable:

```
echo "Los kilometros por hora son: $km"
```

Veamos otro ejemplo completo:

```
#####
```

```
#!/bin/sh
```

```
#Segundo ejemplo, ejecución de un comando
```

```
comando= /etc/X11/xorg.conf
```

```
cat $comando
```

```
#####
```


Bash

Sobreviviendo en GNU/Linux



Para este ejemplo necesitamos permiso de root, ya que es un archivo de configuración que afecta a todo el equipo.

Lo que hace el script es pasar como parámetro el comando cat a la variable `$comando`, cuyo contenido es la dirección del xorg.

Esto no es todo lo que le respecta a las variables, sin embargo hay de dos sopas...XD, se esperan a la segunda parte de esto o tal vez tercera, o lo buscan antes...-)

Paso de argumentos desde la konsola

Como lo menciono en el subtítulo, podemos enviar parámetros desde la konsola, pero para que quede más claro que el agua pasemos al ejemplo práctico:

=====

```
#!/bin/sh
```

```
«Cuarto programa, paso de parámetros desde la konsola
```

```
«Se usa: ./cuarto >nombre>>apellido>
```

=====

```
echo "Bienvenido a $0"
```

```
echo "Tu nombre es: $1"
```

```
echo "Tu apellido es: $2"
```

```
echo "$@"
```

Bash nos da algunas variables predefinidas, en este caso son `$0`, `$1`, `$2`.

Si ejecutamos, no dará una salida como la siguiente:

```
>lyonn@localhost >>$ ./cuarto.sh rodrigo ramos
```

La salida es la siguiente:

```
Bienvenido a ./cuarto.sh
```

```
Tu nombre es: rodrigo
```

```
Tu apellido es: ramos
```

```
rodrigo ramos
```

`$#` Es la cantidad de parámetros pasados a la función.

`$?` Es el valor devuelto del último programa ejecutado dentro del script.

Bash

Sobreviviendo en GNU/Linux



Lectura de datos.

Algunos scripts necesitan que el usuario interactúe con él. Puede ser el caso de un menú con las actividades que con mayor frecuencia realizamos.

La entrada de datos la haremos por medio del teclado con el comando read, veamos un ejemplito antes de empezar con los menús.

```
#!/bin/sh
```

```
#####
```

```
#Quinto ejemplo, comando read
```

```
echo -n "Dame la marca de tu automóvil:"
```

```
read marca
```

```
echo -n "Dame el modelo: "
```

```
read modelo
```

```
echo -n "Dame el color: "
```

```
read color
```

```
echo -n "Su automóvil es un $marca de color $color y el modelo es $modelo"
```

-n hace que cuando se imprime no haya salto de línea.

Para mayor comprensión del ejemplo del menú decidí exponerlo hasta que veamos las estructuras de control o sentencias condicionales.

Operadores.

Los operadores también forman parte de bash, siendo estos de gran importancia para las tareas de mediana a mayor complejidad.

La sintaxis es:

```
$>expresión>
```

```
mivar=$>150*45+2-6/5>
```

Bash

Sobreviviendo en GNU/Linux



Lista de operadores:

Operador	Descripción
+, -	Adición y sustracción
*, /, %	Multiplicación, división y resto
!	Negación
<<, >>	Movimiento a nivel de bits
!=, ==	Desigualdad e igualdad
<=, >=, <, >	Comparaciones de menor o igual a, mayor o igual a, menor a, mayor a
&	<u>and</u>
^	<u>or</u> exclusivo
	<u>or</u>
&&	<u>and</u> lógico
	<u>or</u> lógico
-d	Si el archivo es una carpeta
-f	Si el archivo es común
-r	Si el permiso de ejecución está activado
-s	Si el largo del archivo es mayor que 0
-w	Si el permiso de escritura está activado
-x	Si el permiso de ejecución está activado
-e	Verdadero si el archivo existe
-c	Verdadero si el archivo existe y es un dispositivo de caracteres.

Como saben, a mi me gusta ejemplificar todo, si pasa una mosca lo ejemplifico, si un elefante es azul lo ejemplifico (soy malo contando chistes...5), así que ejemplificaré:



Bash

Sobreviviendo en GNU/Linux



```
#!/bin/sh
```

#Sexto, operación con variables

```
#
echo "Ingresa el primer número"
read num1
echo "Ingresa segundo número"
read num2
res=$((num1*num2+num1+num2))
echo "El resultado es $res"
```

El resultado es como el siguiente:

```
Ingresa el primer número
2
Ingresa segundo número
1
El resultado es 5
```

Este script recibe dos números, hace la operación y arroja el resultado.

Estoy de acuerdo que nunca ocuparemos un script de este tipo, pero es bueno saber lo más básico, así empezaremos a inmiscuirnos cada vez más profundo en bash y en otros lenguajes también.

Sin más preámbulos pasemos a algo que verdaderamente es importante y esto son las estructuras condicionales.

Estructuras condicionales.

Bien, empezaré por el muy conocido if. Esta sentencia quiere decir Si en forma condicional, por ejemplo, si te sales te enfrias.

Sintaxis:

```
if > expresión >
then
    acción1
elif > expresión2 >
then
    acción2
else
    acción3
fi
```

Bash

Sobreviviendo en GNU/Linux



La primera línea evalúa la expresión, si es verdadera ejecuta acción1, si es falsa pasa hasta el else y ejecuta a acción2. Si es verdadera la primera condición, y después de ejecutar a acción1 evalúa expresión2, si es verdadera ejecuta acción2, insisto: cuiden los espacios entre operadores y operandos.

elif es la contracción de else if.

Ejemplo:

```
#!/bin/sh
echo "Ingrese su valor: "
read valor
if > $valor = 200 >
then
    echo "El valor es igual a 200"
else
    echo "El valor es distinto de 200"
fi
```

Una vez más no hace falta explicar que hace este script. Sólo quiero que noten los espacios en la sentencia if.

Veamos otro ejemplo con elif:

```
#!/bin/sh
#Octavo, uso de elif

echo -n " Dame el nombre de un archivo o directorio "
read a
if > -d $a >
then
    echo "Esto es un directorio(carpetas)"

elif > -f $a >
then
    echo "Esto es un archivo"
else
    echo "Esto no se que sea"
fi
```

Bash

Sobreviviendo en GNU/Linux



Nota: En caso de que se introduzcan nombres de carpeta, terminar con /, si es un archivo llevará su extensión. Ejemplo: directorio/archivo.sh Ahora estamos utilizando los operadores -d y -f, los cuales están brevemente explicados en la tabla de operadores.

Case.

La estructura condicional case es muy usada para los menús. Recuerdo que alguna vez me dijeron que un case es un "turbo if".

La sintaxis es:

```
case >variable> in
  opcion1)
funcion1(...);
  opcion2)
funcion2(...);
...
esac
```

Ejemplo:

```
#!/bin/sh
#novenq ejemplo case
echo "1: Ver archivos"
echo "2: Crear directorios"
echo "3: Ver los dispositivos (chipset)"
echo "4: Ver el tiempo que llevo conectado"
echo "5: salir.."
read -p "Ingresar opción " op
case $op in
1)ls;;
2)echo -n "Nombre de la carpeta "
read dir
mkdir $dir;;
3)lspci;;
4)uptime;;
5)exit;;
*)Opción no valida
exit 1;
esac
exit 0
```

Bash

Sobreviviendo en GNU/Linux



Ahora presento el menú, es sumamente sencillo, pero nos da una idea muy clara de lo sencillo que es trabajar con bash. Con esto no quiero decir que bash sólo de para esto, al contrario, es muy amplio.

While (mientras). while es muy usado en los lenguajes de programación, ya que facilita la creación de bucles. While se ejecuta de 0 a muchas veces.

Sintáxis:

```
while >condicion>
do
  accion1
  ...
done
```

Ejemplificando:

```
#!/bin/sh
#decimo, uso de while
op=1
while > $op != 100 >
do
echo "Tu opción es $op "
echo -n "Para salir introduzca 100 "
read op
done
```

Al principio le damos el valor de uno a la variable que contendrá la opción que nosotros elijamos, ya que como mencioné anteriormente, el ciclo while de ejecuta 0 o más veces. Nuestro ciclo no saldrá hasta que no ingresemos el valor de 100.

For (hasta). Este ciclo es muy usado, una ventaja o desventaja es que muchas veces tenemos que saber cuantas iteraciones tiene que hacer.

Sintáxis:

```
for >variable> in >variable>
sentencias
```

Bash

Sobreviviendo en GNU/Linux



Ejemplo:

```
#!/bin/sh
```

```
#undecimo, ejemplo for
```

```
val=10
```

```
for x in 1 2 3 4 5 6 7 8 9
```

```
do
```

```
echo "El valor de x es $x"
```

```
done
```

Este código nos mandará una salida así:

```
El valor de x es 1
```

```
El valor de x es 2
```

```
El valor de x es 3
```

```
El valor de x es 4
```

```
El valor de x es 5
```

```
El valor de x es 6
```

```
El valor de x es 7
```

```
El valor de x es 8
```

```
El valor de x es 9
```

Implementación de nuestras funciones.

Nosotros podemos definir nuestras propias funciones, eso facilita mucho el trabajar con códigos relativamente largos, ya que cuando dividimos en funciones nos será más fácil detectar errores y así mismo corregirlos.

Bash no es la excepción, éste nos permite definir nuestras propias funciones.

Sintaxis:

```
funcion()
```

```
>
```

```
sentencias
```

```
....
```

```
>
```


Bash

Sobreviviendo en GNU/Linux



Las funciones deben aparecer al inicio del script, y si tenemos más de una será una debajo de la otra.

Para llamarlas es:

```
funcion()
```

Dentro de las funciones los parámetros son llamados como si los pasáramos cuando ejecutamos el script desde la konsola (`$0`, `$1`, `$2...`)

Un ejemplo de función de suma es:

```
#!/bin/sh
```

```
# Dodecimo, Suma con paso de parámetros a la función Sumar
```

```
Suma()
```

```
>
```

```
echo -n "Suma "
```

```
c=$((a+b))
```

```
echo "$c"
```

```
>
```

```
echo -n "Ingrese dos valores para sumar "
```

```
echo -n "Valor 1: "
```

```
read a
```

```
echo -n "Valor 2: "
```

```
read b
```

```
Suma $a , $b;>
```

La salida es:

```
Ingrese dos valores para sumar Valor
```

```
1: 3
```

```
Valor 2: 4
```

```
Suma 7
```

Bash

Sobreviviendo en GNU/Linux



Comandos.

Una de las especialidades de bash es la ejecución de comandos como ya vimos en el menú. También podemos ejecutar un programa con exec:

```
exec >programa>
```

Si ejecutamos un exec desde nuestra konsola, este matará el proceso padre, es decir, ejecutará a 'programa' y saldrá de konsola, es como si ejecutáramos con alt+F2 y después ingresamos el comando que queramos.

Ejemplo:

```
#!/bin/sh
```

```
echo -n "Ejecución de exec"
```

```
sleep 4
```

```
exec firefox
```

Esto hace que nos imprima 'Ejecución de exec', después la konsola "duerme" durante 4 segundos, pasando ese tiempo ejecuta nuestro navegador firefox. Para terminar no sale de la konsola.

Nosotros podemos pasar como parámetro el programa queremos que ejecute, eso les toca a ustedes. Diviertanse con lo aprendido hasta ahora, jueguen con lo que han aprendido y pruebenlo.

Nos vemos el próximo número de esta revista y que ojalá haya sido de su agrado. Si tienen una idea o no les pareció cómo lo expliqué haganmelo saber. Se acepta todo tipo de crítica y sugerencias.

```
:-*/ Happy Hacking \*:-
```

xfce

lo que un linuxero deve saber



XFCE

XFCE

Xfce

lo que un linuxero debe saber



Hoy les hablaré del entorno de escritorio Xfce, un entorno de escritorio muy diferente.

¿Por qué Xfce?

Porque es un entorno de escritorio bastante ligero en comparación con GNOME y KDE, lo cual nos permite que cargue de forma rápida el entorno.

Puede ser utilizado en los sistemas basados en Unix, por ejemplo GNU/ Linux, NetBSD, FreeBSD, OpenBSD, MacOS X, Solaris etc.

Lo que debes saber.

El creador de este entorno de escritorio es nada mas y nada menos que Olivier Fourdan. Él describe a Xfce como:

"Xfce es un entorno de escritorio ligero para varios sistemas * NIX. Diseñado para la productividad, se carga y ejecuta aplicaciones rápido, mientras conserva recursos del sistema".

El nombre Xfce originalmente provenia de XForms Common Enviroment, pero debido a los grandes cambios en el código, ya no usa el kit de herramientas de XForms, como originalmente lo hacia. El nombre sobrevivió, pero ya no se indica como XFce sino Xfce. Los desarrolladores están de acuerdo en que el nombre carece de significado actualmente, aunque se le suele desglosar como X Free Cholesterol Environment (entorno X libre de colesterol) en referencia al poco consumo de memoria que realiza y a la velocidad con que se ejecuta al no tener elementos superfluos a diferencia de otros entornos de escritorio más grandes.

Wikipedia

Características:

Basado en GTK

Gestor de ventanas: Xfwm

Gestor de archivos: Thunar

Gestor de archivos comprimidos: Xarchiver

Actualmente Xfce está disponible en su versión 4.1 en su pagina oficial: <http://www.xfce.org/>

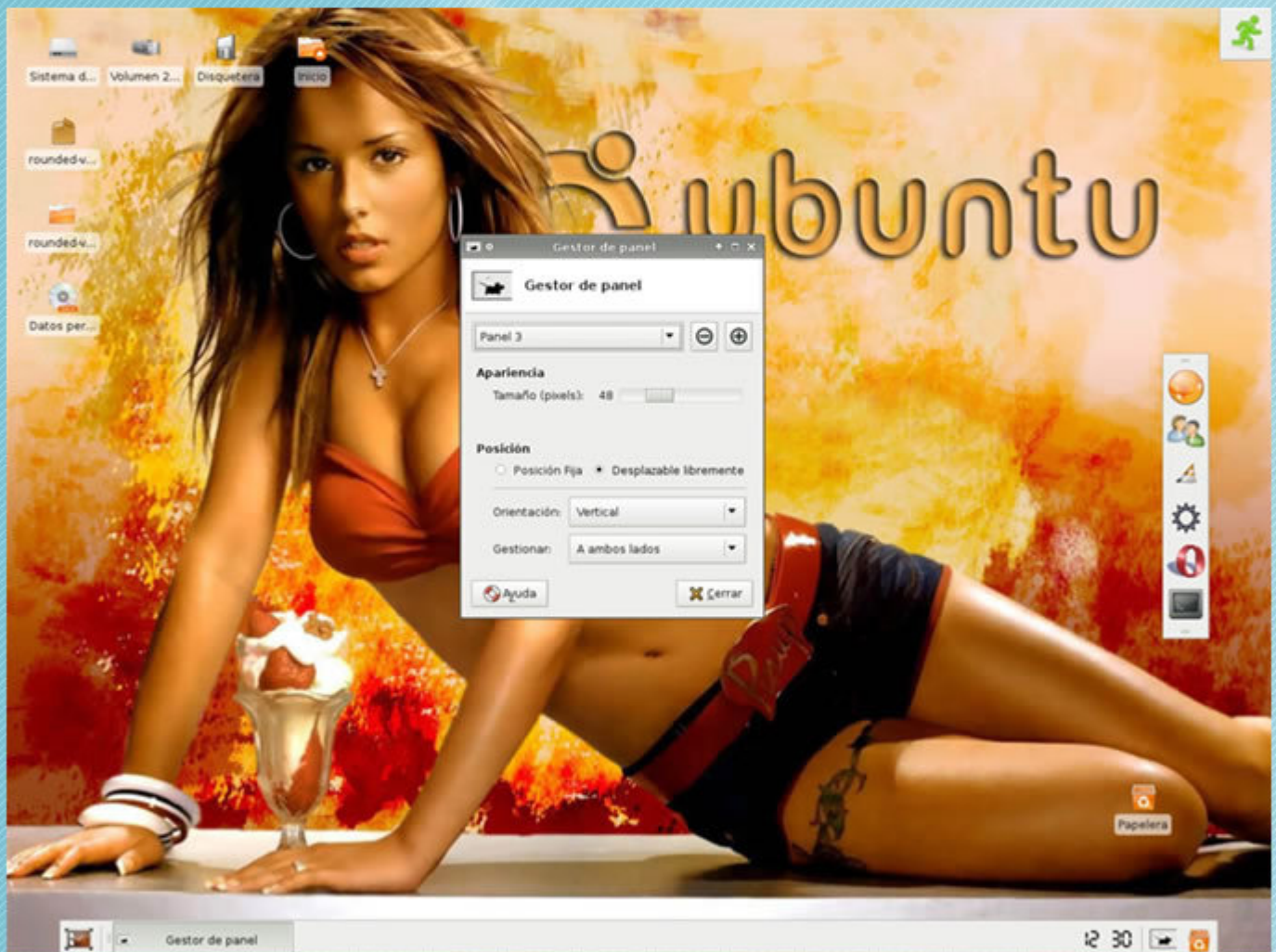
Xfce

lo que un linuxero debe saber



Pantalla inicial

Xfce está conformado por componentes, esa es la razón por la cual es ligero. Esto te permite personalizar el entorno, como agregar o quitar paneles a nuestro gusto y eso en lo personal es lo que más me agrada.



Al inicio solo tenemos un panel vacío, y los iconos principales como son: carpeta de inicio (/home), unidades de disco, sistema de archivos y papelera (Trash). Esto hace que los usuarios nuevos nos desconcertemos un poco, pero no hay problema, para acceder al menú solo basta con dar un click derecho sobre el escritorio.

Xfce

lo que un linuxero debe saber



Para agregar un nuevo panel es muy sencillo, como les había comentado al principio, sólo aparece un panel vacío, podemos ocupar ese y si queremos otro basta con darle un click derecho en alguno de sus extremos y darle click en personalizar panel



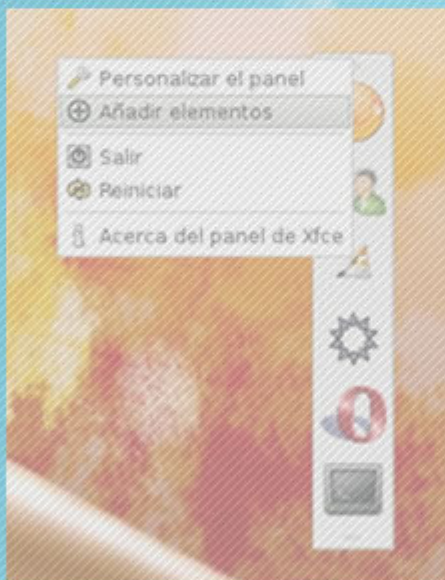
Xfce

lo que un linuxero debe saber



Para agregar un nuevo panel le damos click en el signo de más "+" y para quitar el de menos "-" algo muy sencillo.

Si queremos agregar lanzadores (accesos directos a aplicaciones) o accesorios como la papelerera, mostrar escritorio, reloj, etc. Igualmente damos click derecho en uno de sus extremos y le damos click donde dice "Añadir elementos".



Ejemplo para un lanzador.

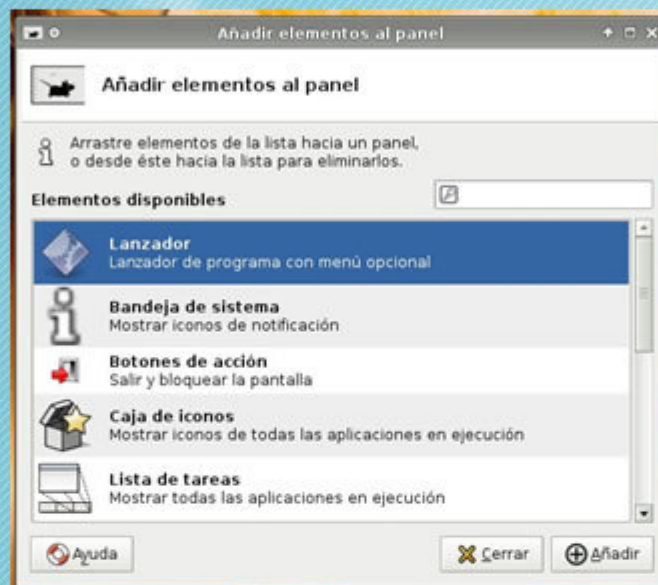


Xfce

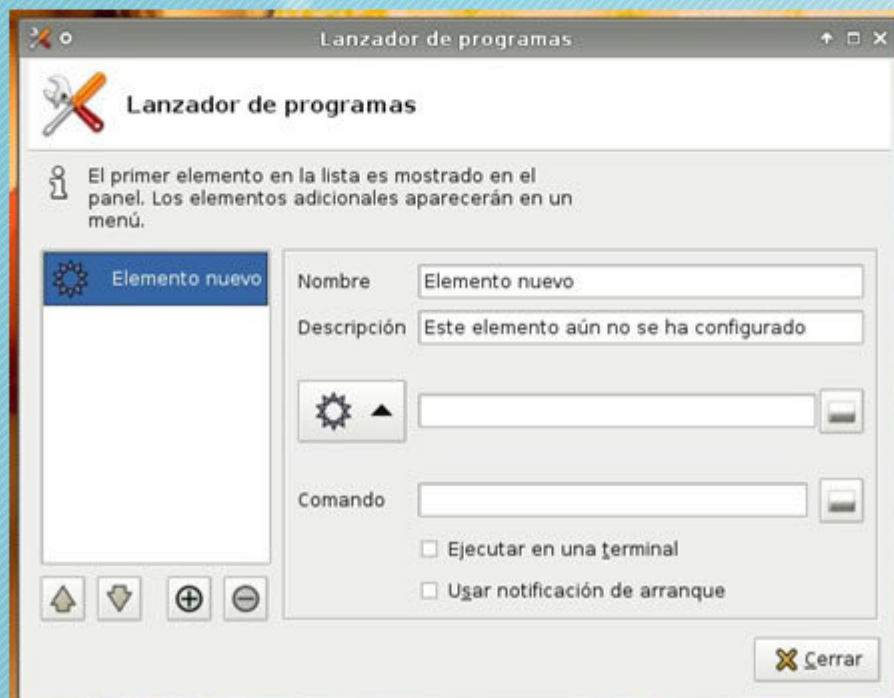
lo que un linuxero debe saber



Seleccionamos la opción de lanzador y le damos click en "Añadir"



Nos mostrará el siguiente cuadro de diálogo donde pondremos el nombre de nuestro lanzador, una descripción que aparecerá cuando posemos el cursor sobre el icono del lanzador, el icono que deseemos que tenga y el comando con el cual se abrirá la aplicación.

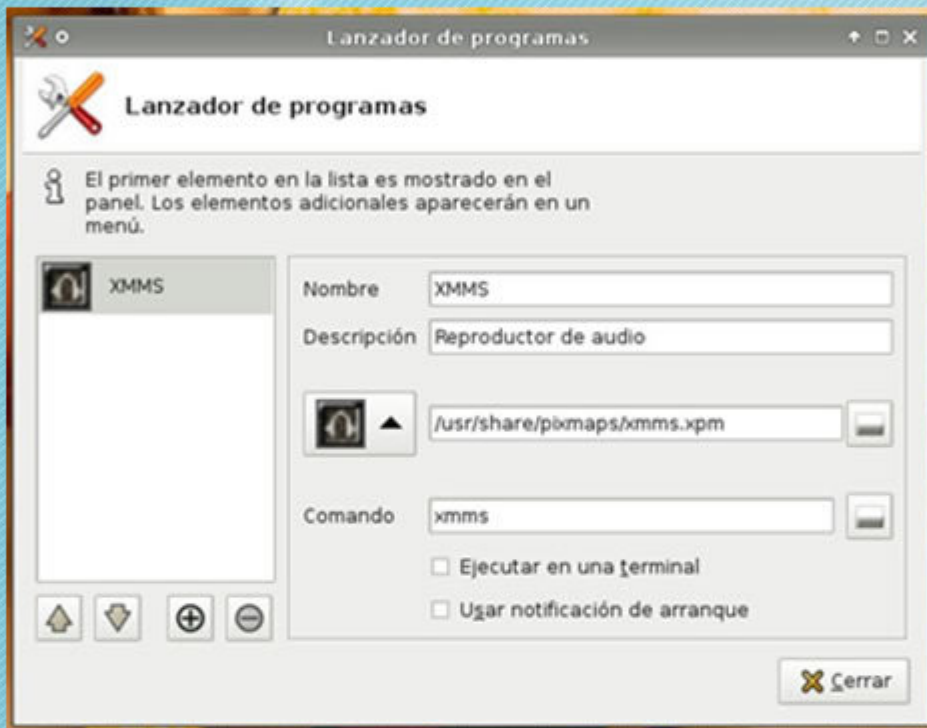


Xfce

lo que un linuxero debe saber



En mi caso hice el lanzador para XMMS

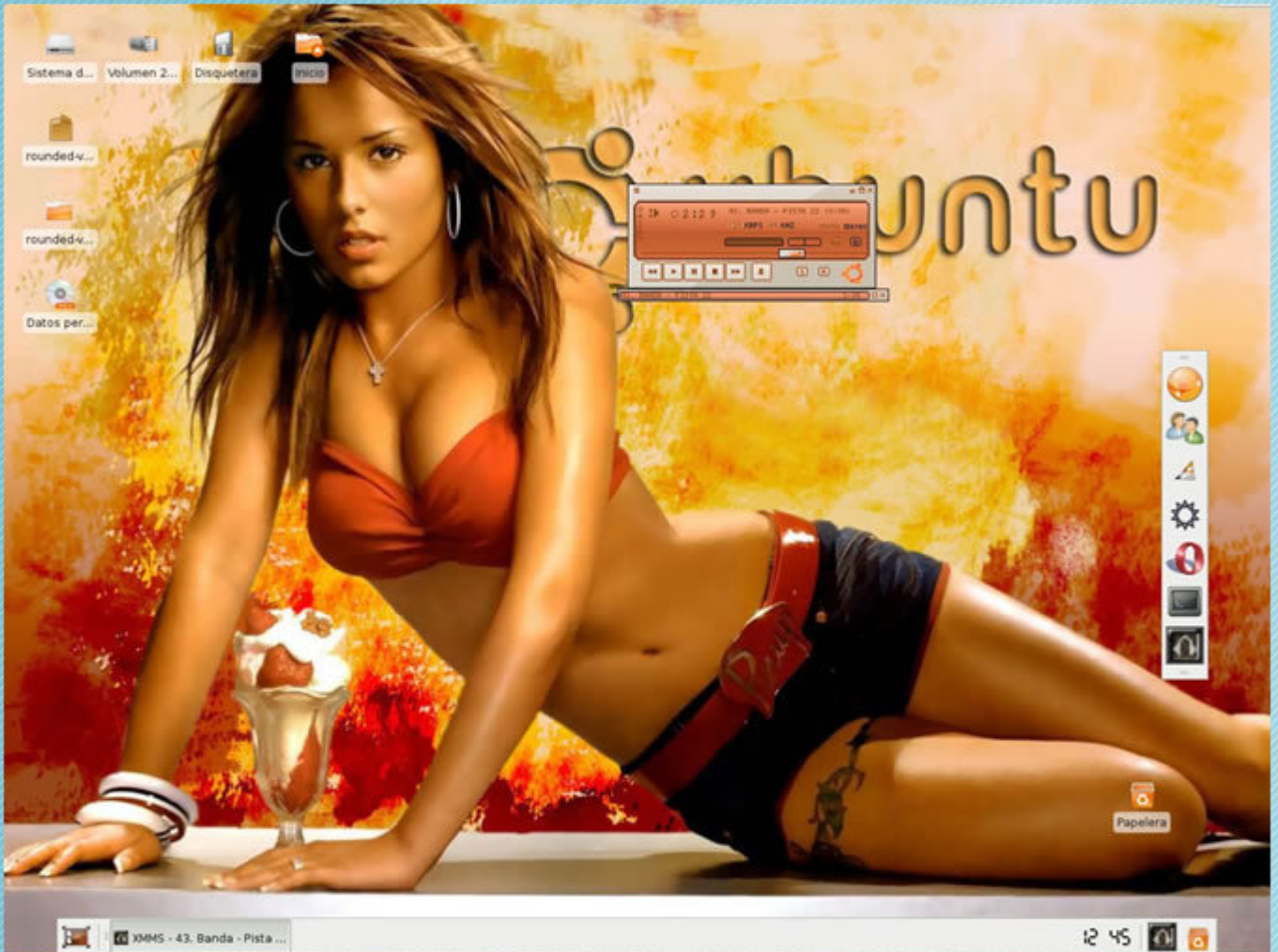


Y le damos click en cerrar, listo!! tenemos nuestro nuevo lanzador =D, a disfrutar de nuestro entorno personalizado.



xfce

lo que un linuxero debe saber



Como pueden ver, la personalización es muy fácil, ya que es un entorno muy intuitivo y fácil de usar para el usuario una vez que explora lo básico.

Para configurar resolución de pantalla, wallpaper, iconos etc. sólo damos click derecho >>Configuración>>Administrador de configuración.

Si tienen Ubuntu y quieren probar xfce lo pueden instalar con:
`sudo apt-get install xfce4`

Un saludo.

A hackear!!!

Fedora

lo que un linuxero debe saber



Fedora

lo que un linuxero debe saber



Hasta que se me hizo hablar de hasta ahora mi distro favorita. Les hablaré de su historia, no es mucho pero considero importante que se conozca acerca de esta distro y no solo Fedora sino las muchas otras tan conocidas como la no tanto.

Fedora, antes conocida como Fedora Core. ¿Por qué cambio a únicamente Fedora? Cambio porque no sólo incluye "lo básico" del sistema operativo (core significa base en español) sino implementa más aplicaciones, con lo cual brinda mayor funcionalidad, ya que no olvidemos que es una distro de propósito general y eso conlleva a mayores opciones para los usuarios novatos y los no tanto. El cambio de nombre surgió hasta la versión 7 (Actual).

Fedora Core es un fork de Red Hat, la cual se dedica actualmente a su versión empresarial y el soporte de la misma. La primera versión de Fedora salió en el 2003, llamándola Yarrow y se han mantenido estos lanzamientos dos veces por año.

Nombre	Versión	Lanzamiento
Yarrow	Core 1	5 de noviembre de 2003
Tettnang	Core 2	19 de mayo de 2004
Heidelberg	Core 3	8 de noviembre de 2004
Stentz	Core 4	13 de junio de 2005
Bordeaux	Core 5	20 de marzo de 2006
Zod	Core 6	24 de octubre de 2006
Moonshine	7	31 de mayo de 2007
WereWolf	8	8 de noviembre de 2007

Fuente: Wikipedia

Como todo, los nombres tienen un porqué, y en este caso no tienen alguna relación.

Fedora

lo que un linuxero debe saber



Nombre	Origen
Yarrow	La Milenrama (Yarrow en inglés) es una planta con usos variados, entre los cuales está el de condimento en la elaboración de la cerveza. Actualmente se reemplaza por el humulus.
Tettnang	Ciudad alemana en que se produce humulus.
Heidelberg	Ciudad alemana que produce y distribuye cerveza.
Stenz	Viticultor francés que producía una serie de bebidas alcohólicas.
Bordeaux	Región francesa productora de vinos. También es un personaje de un libro de cómics.
Zod	Personaje de libro de cómics, enemigo de Superman.
Moonshine	Nombre coloquial que se le da en inglés a las bebidas alcohólicas destiladas de forma casera.
WereWolf	Hombre que se puede convertir en un lobo.

Instalación de paquetes.

Para instalar paquetes existen los tarballs, RPMs y el manejador, entre otros, entiéndase como paquetes programas o librerías

Como manejador de paquetes Fedora usa YUM, el cual es en modo consola y en forma gráfica puedes tener YUMEX o YUM Extendido: o puedes usar pirut.

Fedora

lo que un linuxero debe saber



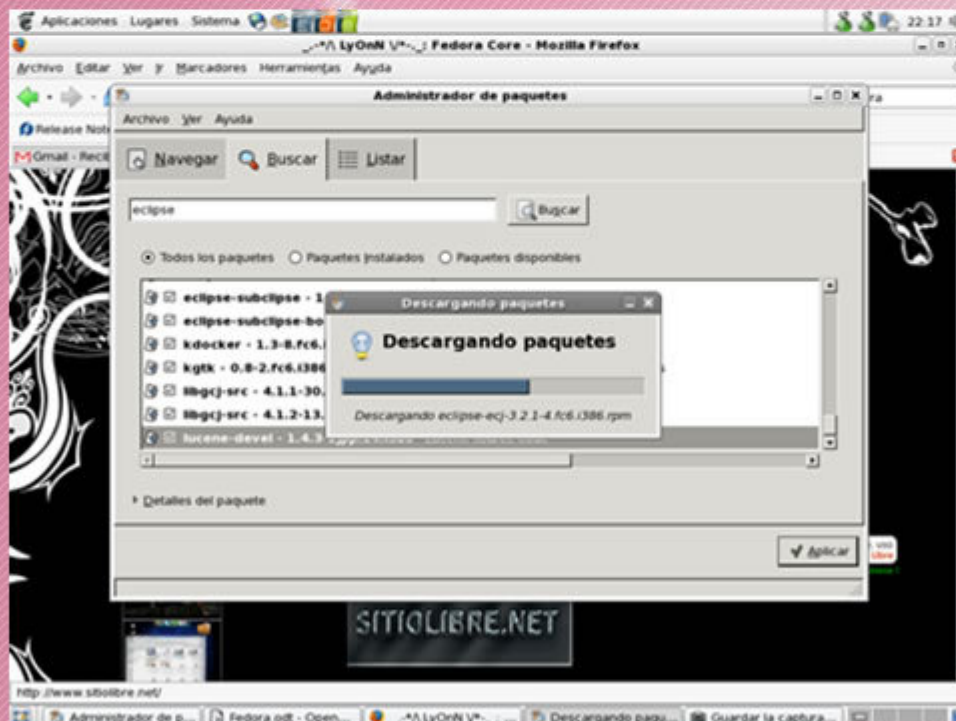
YUM significa Yellow dog Updater Modified, actualmente lo mantiene la comunidad del proyecto DUKE, donde por cierto, está su creador Seth Vidal. Esta comunidad reescribió a YUP (Yellowdog Updater) que puede decirse que es el papá de YUM.

Como sabrán, los paquetes precompilados que Fedora usa son los RPM o Redhat Packet Manager y para instalarlos hay una forma muy simple.

- 1: Doble click sobre el paquete y seguir las instrucciones
- 2: En konsola te sitúas donde está el paquete y escribes.
`$ rpm -ivh paquete.rpm`

Para desinstalar en konsola basta con :

`$ rpm -e paquete`



Ejecutando Pirut en Fedora Core 6

Fedora

lo que un linuxero debe saber



Repositorios.

Fedora tiene por defecto tres repositorios, los cuales no satisfacen todas las necesidades del usuario.

Core--> Paquetes esenciales

Extras--> Paquetes más usados

Updates--> Actualizaciones del sistema

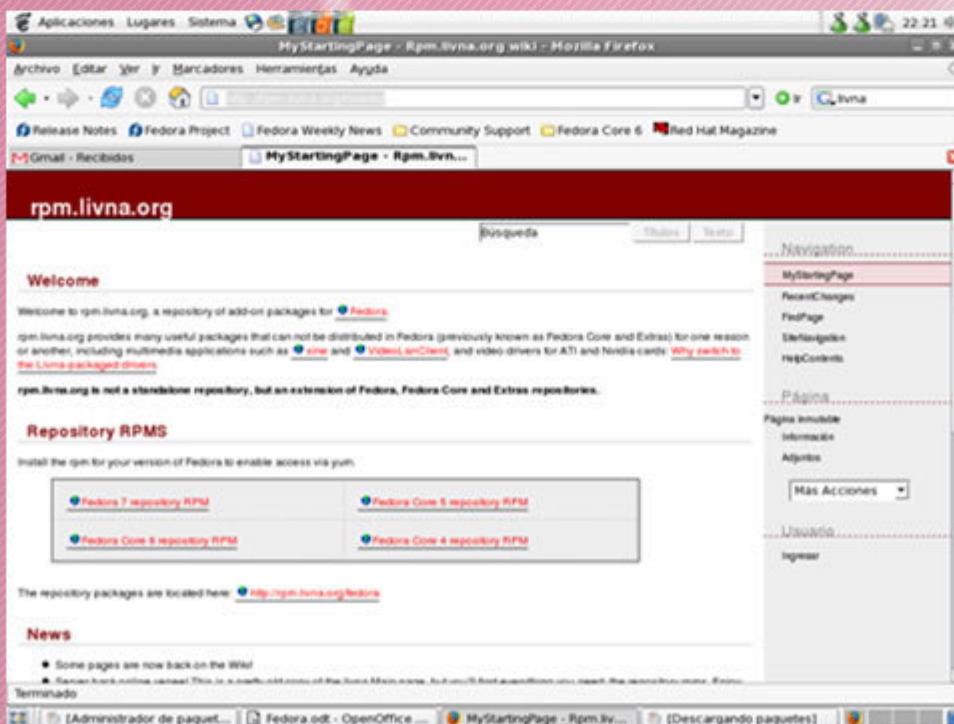
Otros repositorios muy usados, pero no oficiales son los de livna, freshRPM, por mencionar algunos.

Yo he trabajado con livna y me parece muy bueno. Algunos dicen que no es del todo confiable usar repos no oficiales, pero lo que es un hecho es que muchas veces se necesitan para agilizar el trabajo.

Para usar livna solo basta con instalar el RPM. La página oficial de livna es: <http://rpm.livna.org/rlowiki/>
 Por ejemplo, para mi Fedora Core 6 instalé livna desde konsola y como root:

```
# rpm -ivh http://rpm.livna.org/livna-release-6.rpm
```

Con la instalación de livna tendrán oportunidad de escuchar sus .mp3, por ejemplo, ya que cuando bajamos amarok u cualquier reproductor también baja las librerías necesarias para este formato.



Wiki de livna

Open Suse

La distro del Mes...



Open Suse

La distro del Mes...



La última edición de la iguana (ahora más verde que nunca) ha sido de las más esperadas en la comunidad linuxera.

Sin duda, Open SuSE es de las distribuciones más seguras, estables y eficientes de la actualidad.

Al final de este artículo podrán encontrar tres mini guías básicas y elementales para dejar su Open SuSE 10.3 con soporte multimedia, y Compiz Fusion funcionando al máximo nivel.

Características

Kernel	2.6.22-5 (Utiliza glibc 2.6.1 y el compilador gcc 4.2.1)
Ambientes de escritorio	KDE 3.5.7 Previa de KDE 4 GNOME 2.20
Xorg	7.2
Soporte de drivers para tarjetas	Nvidia, Ati, Intel
Soporte para aceleración gráfica	XGLS, AXGL
Soporte formatos restringidos multimedia	MP3
Fácil instalación	1-Click-Install

Como ya han visto, hay muchas novedades, entre las cuales se destaca la instalación fácil de utilidades como drivers y efectos 3D mediante el sistema "1-Click-Install".

Sin duda, ya era hora de que la gente de Open SuSE se preocuparan de hacer su sistema más sencillo de administrar.

La utilización de YaSt en KDE es sin duda una gran ventaja, claro está que YaSt existe desde mucho antes. Ya estaba incluido desde la edición 10.2, sin embargo, he encontrado a esta herramienta de administración como una de las mejores de entre muchas que ya he visto en otras distribuciones de GNU/ Linux.

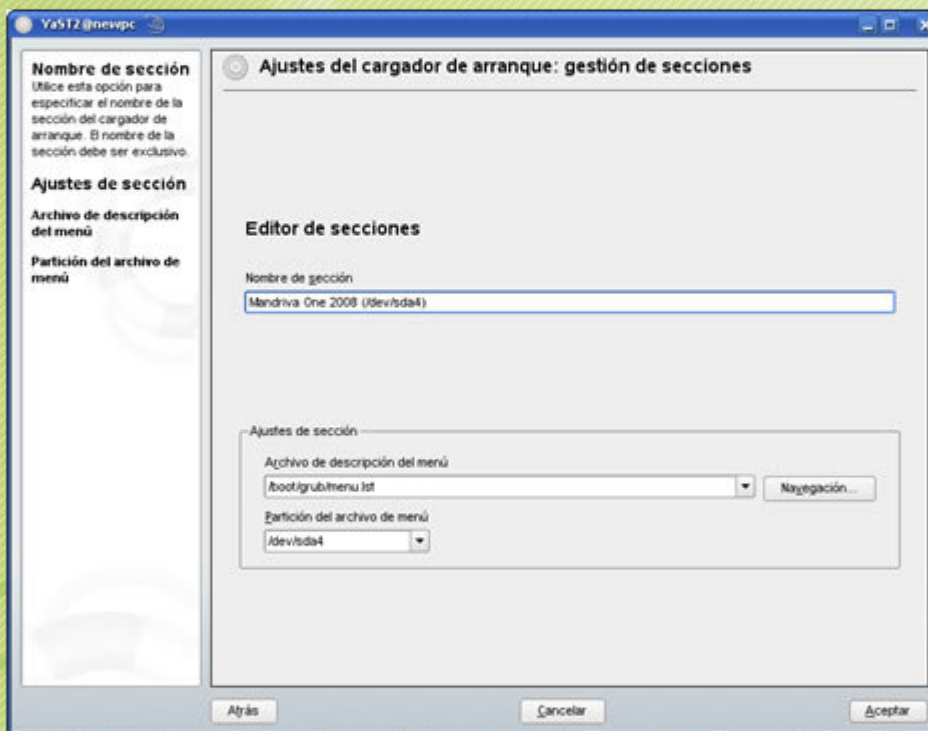
Open Suse

La distro del Mes...



Una de las herramientas que esta herramienta provee es la "Configuración del cargador de arranque" (Grub o Lilo). Ya no es necesario estar editando a mano el fichero que administra el Grub. Ahora basta con escoger el nombre que deseamos para la partición que poseemos y YaSt se encargará de configurar y actualizar el cargador de arranque.

Más sencillo no puede ser...



Finalmente destaco el gran avance en lo que respecta a las actualizaciones. En la edición 10.2 se necesitaba ser un verdadero experto para instalar una actualización, ya que el gestor de actualizaciones era sumamente complejo, complicado y usualmente daba problemas de dependencias.

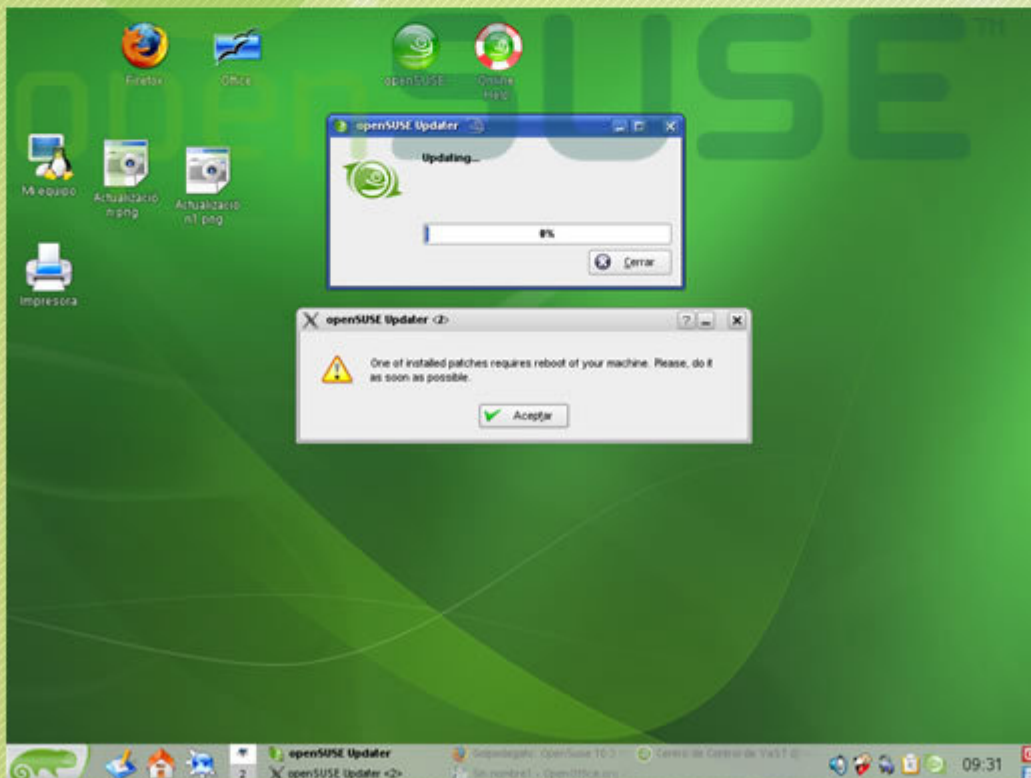
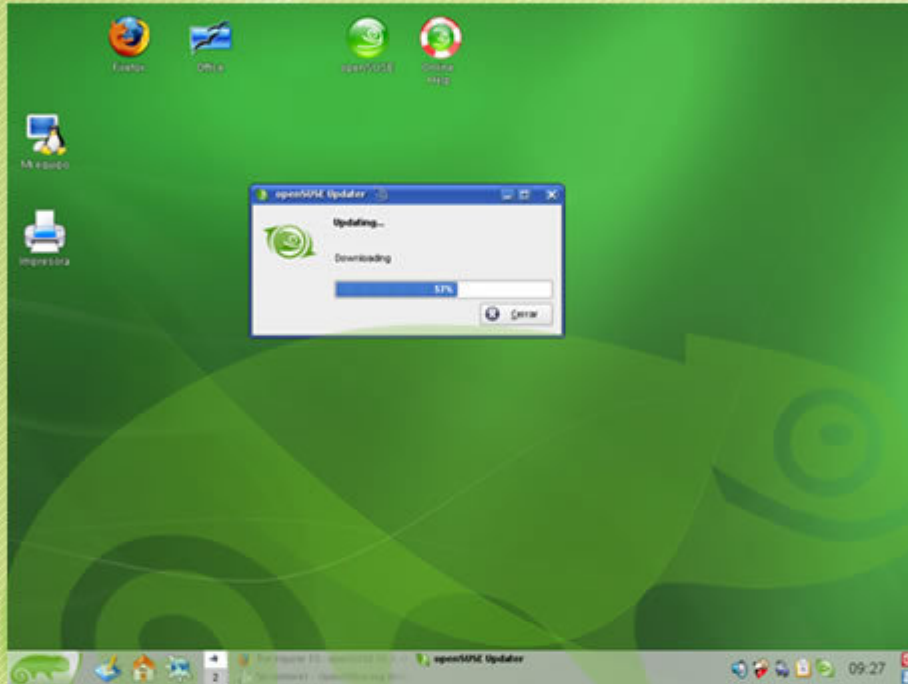
Hoy, la situación es diferente, gracias a OpenSuse Updater la actualización es tan sencilla como en Ubuntu.

Open Suse

La distro del Mes...



Ver para creer:



Open Suse

La distro del Mes...



1-Click-Install

Según noctuido de Fentlinux:

Tal y como su nombre indica, podremos incluir paquetes a un solo click, sin tenernos que preocupar tanto de las resoluciones de dependencias, ya que aunque hay miles de paquetes estos suelen provenir de sitios oficiales como los de Open SuSE como de sitios como Packman project, o Guru's rpm site entre otros. Con esta herramienta además de restar tiempo, restaremos supuestos problemas entre dependencias.

El único inconveniente que le he notado a este gestor de paquetes es la lentitud con la que corre. Aunque no es en realidad un problema de la aplicación por sí misma, sino de la cantidad de funciones que tiene que llevar a cabo para poder instalar los paquetes.

Agregar repositorios, importar las llaves, actualizar los repositorios, buscar el archivo, descarga, instalación y finalmente auto configuración de los paquetes.

Instalar soporte multimedia de formatos restringidos

Flash

Java

Amarok (con soporte MP3) para KDE, o Helix-Banshee para GNOME

DVD encriptados (libdvdcss)

Codecs Extra xine, para MPEG-4 etc. (libxinel)

K3b con soporte MP3 (k3b-codecs)

Win 32 Codecs (w32codecall)

KDE

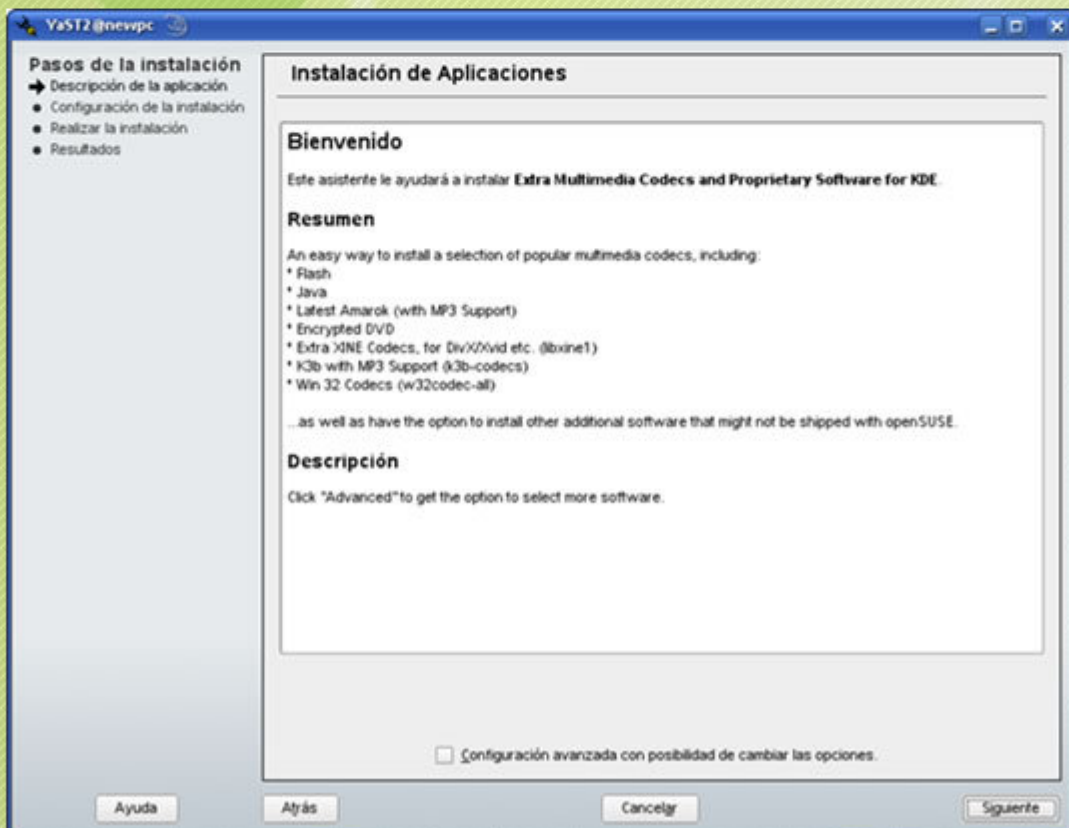
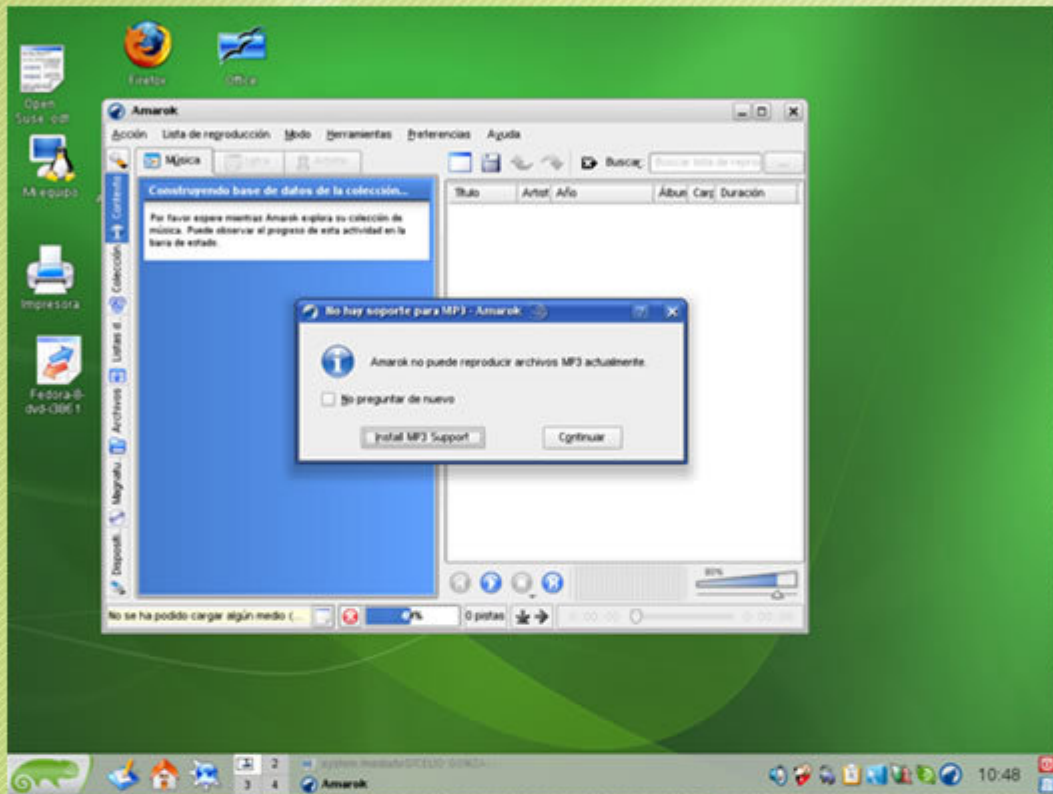
<http://opensuse-community.org/codecs-kde.ymp>

Gnome

<http://opensuse-community.org/codecs-gnome.ymp>

Open Suse

La distro del Mes...



Open Suse

La distro del Mes...



Instalar los drivers para tarjetas Nvidia
Nuevas tarjetas Nvidia

<http://opensuse-community.org/nvidia.ymp>

Tarjetas Nvidia Legacy

<http://opensuse-community.org/nvidia-legacy.ymp>

Instalar Compiz Fusion

KDE

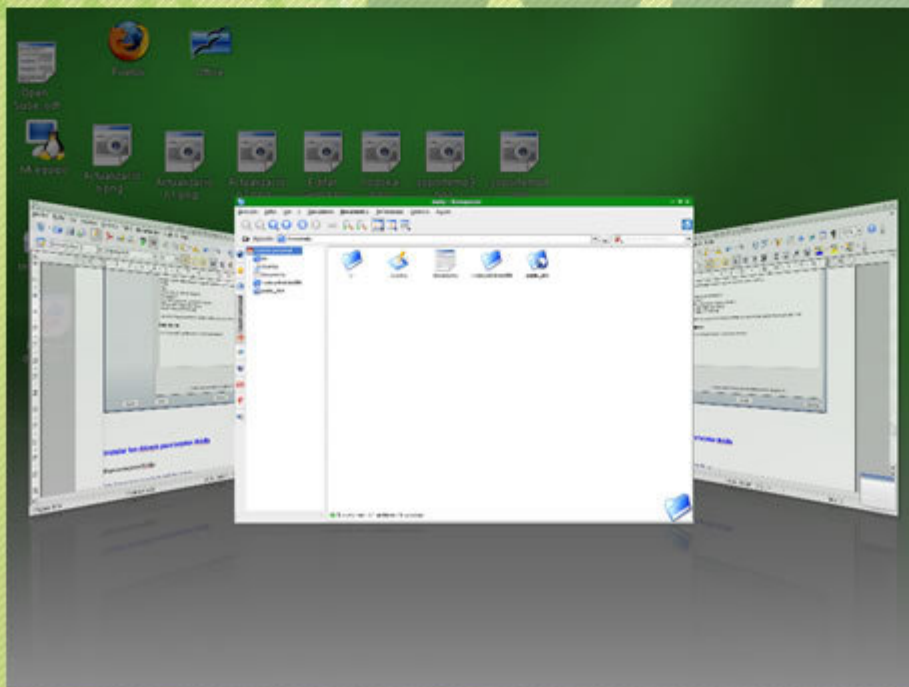
http://download.opensuse.org/repositories/X11:/XGL/openSUSE_10.3/compiz-fusion-kde.ymp

Gnome

http://download.opensuse.org/repositories/X11:/XGL/openSUSE_10.3/compiz-fusion-gnome.ymp

Ícono Compiz Manager

http://download.opensuse.org/repositories/X11:/XGL/openSUSE_10.3/compiz-manager.ymp



Clitec



clitec.org



Cli-Tec y Free-Tux
agradecen su seguimiento,
a ésta, su revista.

<http://clitec.org/freetux>

Espacio Linux

El sitio destacado.

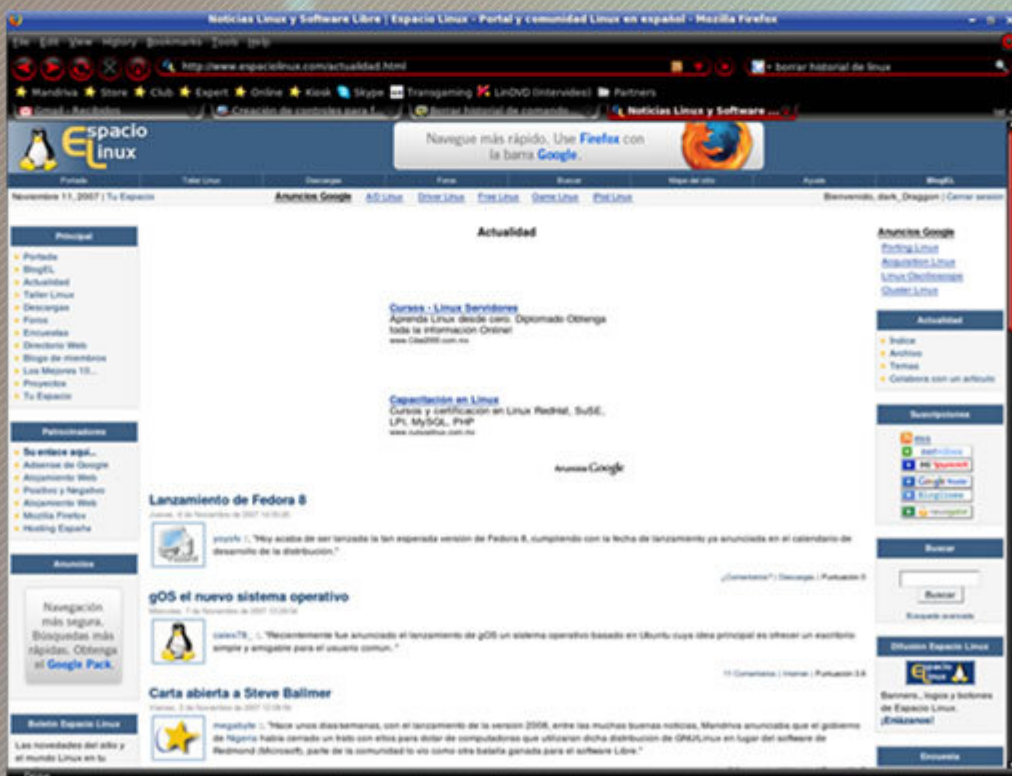


En esta ocasión le damos el sitio del mes a espaciolinux.com (<http://www.espaciolinux.com/>). Espaciolinux es un sitio de ayuda dedicado al software libre, apoyado por muchos y muchos miembros que intercambian sus conocimientos para ayudar a los que apenas inician en el uso de esta plataforma.

Cabe destacar que no sólo es GNU/ Linux, como su nombre sugiere, sino con programación, entornos de escritorio, administración, software, hardware, instalación y muchos otros rubros.

El sitio no es sólo un foro, también contiene noticias, apoyo hacia proyectos (gracias por apoyarnos), talleres, descargas y otras cosas más.

Yo me quedo con el foro, siendo este muy útil y variado, pero el secreto que veo es la dinámica con la que este foro actúa y la velocidad de respuesta de los miembros, así como los conocimientos que estos poseen.



Penguin Protection

Sistemas de Seguridad: Firewall's



Comesaremos con algo sencillo un articulo que esta escrito en mi sitio de seguridad informatica (un poco abandonada) y esta a cargo de Zk un compañero que esta al pendiente de la seguridad en linux.

Una vez sabiendo cuales son los servicios con los que dispone nuestro sistema, entonces podremos decidir cuales de ellos son accesibles y cuales no. Bueno en realidad el problema es un poco mas complicado ya que cuando nuestro sistema se encuentra en medio de dos redes tendremos que decidir desde que red sera accesible un servidor y desde cual no. Pero eso mas que nada es problema de planificación, por lo que tendremos que pensar un poquito mas y estructurarlo tranquilamente (siempre ayuda el papel y el lápiz =0).

Una de las herramientas de seguridad que podemos utilizar son los famosos firewalls que no es nada mas que un sistema que nos permite definir reglas a evaluar cuando un paquete de red pasa por nuestro host. Ya hablando de paquetes que están en nuestro host hay tres situaciones diferentes que podemos evaluar .

1. Un paquete que sale de nuestra maquina y que sale hacia otra
2. Un paquete externo que viene desde otra maquina, pero tiene un destino que no es nuestra maquina.
3. Un paquete de origen externo que viene de otra maquina con destino ala nuestra.

Entonces lo que nos permite hacer un firewall es abrir o no las puertas, la salida o llegada de estos paquetes. Por ejemplo una de las reglas mas tipicas de los firewalls escrita con simples palabras es la siguiente " si llega un paquete desde internet con destino al puerto 80 de mi maquina, entonces, no dejarlo pasar".

Ahora bien lo que tenemos que hacer es adecuar las reglas necesarias de firewall a nuestro sistema. Hay muchas formas, técnicas y combinaciones posibles de instalar un firewall.

Ahora veremos IPTABLES que como su nombre nos dice trabaja con tablas, y estas tablas trabajan con unas cadenas de reglas echas por nosotros.

Penguin Protection

Sistemas de Seguridad: Firewall's



Hay tres cadenas INPUT, OUTPUT y FORWARD, estas son cadenas para definir reglas para lo que ingresa, lo que sale y lo que pasa por nuestro sistema.

Esta es una forma básica del comando iptables.

```
iptables >definición> >cadena> >opción> >acción>
```

El parámetro >definición> es un parámetro general, además que es seguido de las especificaciones de cadena. En esta parte hay varias opciones posibles como :

- A > para agregar una nueva regla a la cadena
- D > para borrar la regla
- L > para visualizar la lista de reglas definidas

En el parámetro >opciones> es donde definimos los filtros.

```
todo lo que sale a internet
todo lo que viene desde la ip 200133.145.12 hacia mi maquina:80
todo lo que sale de mi maquina hacia la ip 200133.145.1
```

En la parte de >acción> se pone lo que tiene que hacer cuando una regla se cumple, ya sea permitirle o rechazarla.

Vamos a ver unos cuantos ejemplos para facilitar esto.

```
iptables -A INPUT -tcp dport 80 -j ACCEPT
```

Con este ejemplo creamos una regla con >-A> para la cadena >INPUT> en la cual definimos que todo lo que entre por el puerto >80> del protocolo >TCP> sea >-p tcp dport > aceptado >-j ACCEPT>.

ojo: siempre requiere el parámetro -p para saber a que protocolo referirse la regla.

```
iptables -A INPUT -p icmp -j DROP
```

Penguin Protection

Sistemas de Seguridad: Firewall's



Con este ejemplo filtramos y descartamos todo lo que ingresa por el protocolo ICMP a nuestra maquina.

```
iptables -A OUTPUT -p tcp -d 192.168.444 -j REJECT
```

En este ejemplo en lugar de definir una regla de firewall para todo lo que entra, lo hicimos para lo que sale de nuestra maquina y se dirige a la ip 192.168.444 ,

esta acción >REJECT> rechaza dicho paquete y como no definimos ningún puerto rechazara todo.

Podemos utilizar una política de funcionamiento, que consiste en en rechazar todo lo que entra y después crear nuestras reglas excepcionales para dicha política.

```
iptables -P INPUT DENY -> denegamos TODO
```

```
iptables -A INPUT -p tcp dport 23 -j ACCEPT ->abrimos lo que necesitamos
```

```
iptables -A INPUT -p tcp dport 80 -j ACCEPT
```

De esta forma tenemos nuestro sistema con unos pocos puertos abiertos.

- > -A (crear regla)
- > -D (Borramos regla)
- > INPUT (ingreso)
- > OUTPUT (egreso)
- > FORWARD (paso)
- > -p (definimos protocolo)
- > dport (definimos puerto)
- > -s (definimos fuente)
- > -d (destino)
- > -i (dispositivo de red)
- > ACCEPT (aceptar)
- > DROP (descartar)
- > REJECT (rechazar)

Penguin Protection

Sistemas de Seguridad: Firewall's



`iptables -L` -> vemos el listado de las reglas que tenemos definidas.

`iptables -F` -> hacemos un borrado a todas las listas definidas.

Para guardar las reglas que definimos (porque una vez apagada la maquina se pierden todas las reglas definidas y que flojera hacer las de nuevo no?).

manera sencilla y restablecerlas diariamente.

`cd /sbin` > directorio de archivos de iptables

`iptables-save -c` > nombre _del_ archivo > nombre que quieras

`iptables-restore nombre _del_ archivo` > creo que esta explicito

Si queremos que cada vez que iniciemos la maquina sola se restablezcan las reglas definidas solo has de hacer un script.

por: zerialkiller (ZK) (www.kernelpanic.org.mx) (gndx)

Richard Stallman



Este lunes 12 de noviembre el equipo de Cli-Tec, o al menos los que vivimos cerca de la Ciudad de México, acudimos a este simposio que organizaron en la UNAM, entre ellos Odin Mujica.

La cita era a las 11:30 a.m en la facultad de ciencias, a la cual llegamos casi puntuales, eso nos permitió sentarnos en un buen lugar.

Más tarde Stallman salió y empezó a regalar stickers, (que por cierto no alcancé :S), de un momento a otro se abalanzaron todos peleando por encontrar algo.

Unos minutos después comenzó a dar su platica sobre el software libre, la cuál está titulada como "El Movimiento del Software Libre"

De tanto en tanto, Stallman hacia algunas bromas con respecto al software privativo, que para ser sincero, si eran graciosas =)...



[iz. A der] Marco, Zion, Richard, Lyonn, Samael.

Richard Stallman

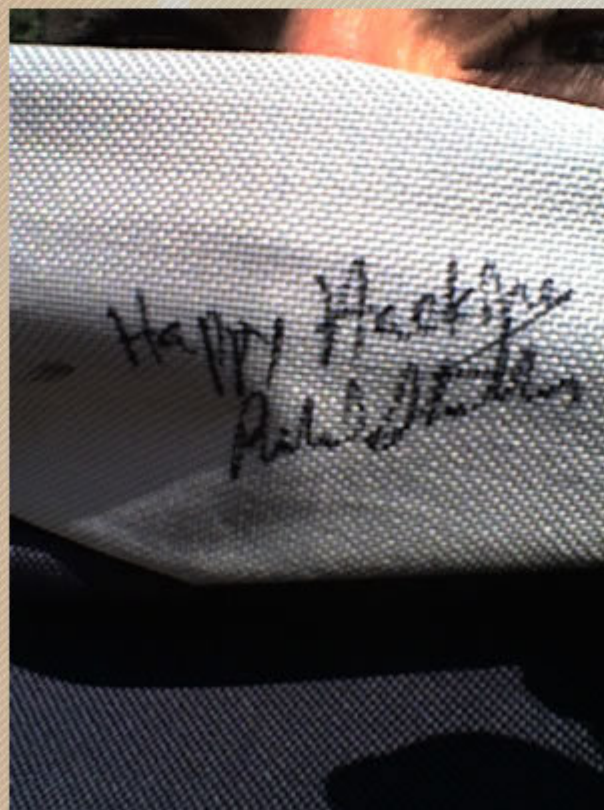


En esta conferencia habló sobre las 4 libertades que el software libre provee, mencionando y haciendo hincapié en que si alguna de ellas falta No es software libre.

Otro aspecto que me gustó es la forma en que maneja a conversación, aunque su español aún es deficiente él es bastante claro y bastante marcado su estilo de pensar, ya que es una persona difícil y eso no es nada nuevo porque se dice que viaja con su colchón a todas partes y no se queda en hoteles.

Cuando acabó la conferencia empezó la hora de las preguntas, las cuales algunas fueron algo fuera de tema y otras interesantes, ahí es donde se vió una de sus grandes cualidades y esa es la de ser muy preciso y contundente con los términos utilizados. Pasando a la hora de los autógrafos me forme para que me firmara, pero resulta que debía pagar 50 pesos mexicanos por una de esas, después descubri que si compraba un botón del mismo precio la firma era gratis, me pregunto por qué no lo supe antes, pero el chiste es que tengo una súper firma en mi mochila de Java y Solaris. jejeje...dice.

Happy Hacking Richard Stallman o algo así.



Richard Stallman



Un aspecto curioso es que el presentador de la conferencia a la hora de dar a notar algunas personas como las autoridades de la UNAM que hicieron eso posible, a unos chavos de Torreón y a nosotrosiiiiii los de Tecámaci!!!!!!!!!!!!!!!!!!!!

Como dije, al final nos formamos, pero en ese lapso pasamos a saludar a Odin, Osvaldo y nos encontramos a jam (integrante de CLI-Tec) y algunas otras personas que conocemos del FLISOL 2007

No podía faltar San iGNUcio de la iglesia de Emacs, el cual bendijo a las computadoras presentes. Para quienes no sepan quien es San iGNUcio les menciono que es solo una parodia que él hace y juega con ella, digamos que es su segunda personalidad.



Richard autografiando el libro de la FSF

Richard Stallman



Richard como: San iGNUcio de la iglesia emacs =P

Después de esta charla tan amena que ofreció Stallman, fuimos a un pequeño receso para que un poco más tarde estuviéramos presentes en la facultad de filosofía y letras donde Federico Heinz (fundador de via libre) y Richard Stallman hicieron acto de presencia.

Federico Heinz nos platicó acerca del Software libre de una manera político-social, el como nos ayuda como sociedad, ya que el Software Libre también es una forma de vida.

Nos platicaba como los monopolios de Software privativo tiene "esclavizados" a los usuarios y la forma de liberarse es acercándose al Software Libre.

Richard Stallman



Federico Heinz en la facultad de filosofía y letras de la UNAM

Hizo una parábola muy interesante, donde él nos comentaba que de nada te sirve algo que no sabes como funciona, de que está hecho y como lo puede hacer, esto se ve reflejado en los monopolios de software, los cuales te venden software pero de que te sirve tenerlo, sino puedes saber como funciona, no puedes saber que hace en "realidad" internamente, en pocas palabras, no lo puedes estudiar.

Y por ello la alternativa a esto es el camino del Software Libre donde sabemos que tenemos siempre las 4 libertades.



Lynn, Federico Heinz y Marco

Richard Stallman



Después de terminada la charla de Heinz prosiguió Richard ahora con el tema de "Patentes informáticas".

En la cual hizo referencia a como las patentes atentan contra los desarrolladores, obligándolos a pagar por la patente o violarla con el riesgo que eso implica. Trató de explicarnos como los dueños de las patentes buscan hacerlas lo más simples posibles para poder abarcar más terreno.

Es por eso que es muy fácil violar una patente, por las cuales se tiene que pagar y Richard por supuesto está en contra de esto y nos comentaba que habrá que hacer una campaña de protesta contra dichas patentes.

Al igual que la primera plática, hizo uso de su buen humor y al final le pedimos que cantara, pero no quiso =(.... así que solo puso una grabación que tenía en su laptop =).

Al final de nuevo se hicieron presentes las fotos y autógrafos.

Todos los asistentes quedaron muy complacidos ya que siempre al final de las charlas se daba un tiempo para las preguntas lo cual aclaraba muchas dudas.

Esperamos que se sigan haciendo charlas de este tipo más seguido y que más gente se sume a este movimiento que como ya hemos dicho antes "Apenas comienza".



Richard Stallman en la facultad de filosofía y letras



1º de Diciembre



Día de la lucha mundial Contra
el Sida.

Contacto



Clitec

<http://clitec.org>

<Http://clitec.org/freetux>

Miembros Clitec

Zion > Jauc27@gmail.com

lyonn > Crackduoro@gmail.com

Shadow > Angelisaiasga@hotmail.com

GnDx > my@gndx.org

Jam1138 > jam1138@gmail.com

Colaboradores Extranjeros

Menroot > Menroot@gmail.com

Rodolfo > Rodolfoforfq@hotmail.com

MAty > maty1206@gmail.com