



EL CONSULTORIO DE KLAUS



Klaus Knopper es el creador de Knoppix y co-fundador de la LinuxTag Expo. Trabaja en la actualidad como profesor, programador y consultor. Si tiene algún problema de configuración, o simplemente quiere conocer mejor cómo funciona Linux, no dude en escribir sus preguntas a: preguntas@linux-magazine.es.

Hace dos meses un lector me escribió algo como “tengo entendido que un sistema Linux puede “llenarse” cuando se queda sin inodos, cuando se crean demasiados archivos pequeños”. El lector tenía dos preguntas acerca de los números inodos:

- ¿Qué pasa con el número inodo asignado a un archivo que ha sido borrado?
- ¿Se vuelve a asignar a archivos de nueva creación?

Le di una respuesta correcta pero muy técnica a esta pregunta de la creación de

nuevos números inodo. Pero tras una reflexión posterior, se me ocurrió que tal vez lo que el lector quería saber era “qué condiciones pueden llevarnos a un “sistema de archivos lleno” debido a que no quedan más números inodo”, y cómo podía solucionarse.

Puede resultar una situación sorprendente y molesta cuando una partición aparece casi vacía con el comando *df*, lo mismo que cuando cada vez que queremos escribir un archivo aparece el mensaje “no queda espacio en disco”.

Esto es fácilmente reproducible creando un disco floppy con un sistema DOS o ext2, y si a continuación escribimos muchos archivos pequeños hasta que aparece este mensaje misterioso que nos dice que no queda espacio en disco, aunque podemos ver que realmente nuestro ordenador nos está mintiendo. Consideremos este pequeño script de shell Bash como ejemplo. El script hace un formateado rápido a un floppy DOS, lo monta como */media/fd0*, y entonces trata de llenar el espacio disponible con multitud de pequeños archivos:

```
mkdosfs /dev/fd0
mount -t vfat /dev/fd0 >
/media/fd0
count=0; while true; do
echo "What a waste of space" \
> /media/fd0/f${((count++))}
.txt || break
done
```

Tras ejecutar este script, podemos verificar el espacio ocupado en el floppy con:

```
df /media/fd0
```

El hecho de que haya aún espacio disponible, aunque sea imposible usar este espacio para añadir nuevos archivos, no es una limitación exclusiva de DOS. Si repetimos el experimento con ext2 en lugar de (v)fat, obtendremos el mismo resultado.

La razón es que la mayoría de los sistemas de archivos (los más viejos, y también los más estables) usan una especie de array estático para su índice de datos y la lista de archivos y directorios, y su tamaño se determina en el momento de creación del sistema de archivos (al llamar a *mkdosfs* o *mke2fs*). Para ext2 podemos especificar fácilmente el tamaño del índice de archivos (o, en términos Unix, el “número de inodos” que puede contener el sistema de archivos) usando la opción *-N* de *mke2fs*, o la opción *-r* de *mkdosfs*. Pero no podemos cambiar este tamaño fijo de la cabecera del sistema de archivos con posterioridad (aunque es posible cambiar el tamaño de la parte de “información” de una partición ext2 con *resize2fs* tras cambiar el tamaño de la partición con *fdisk*).

Si borramos un archivo en un sistema de archivos que ha usado todos sus números de inodo, vuelve a quedar uno disponible para un nuevo archivo o directorio. El número del inodo del nuevo archivo será el mismo dependiendo de la implementación del sistema de archivos.

La mayoría de las distros usan una tabla estática de tamaño medio para el índice de archivos con ext2 con objeto de permitir un número suficiente de archivos/inodos sin ocupar mucho espacio con entradas vacías. Si planeamos tener un número muy grande de pequeños archivos, deberíamos incrementar el número de inodos, como con la opción *-N* de ext2. Pero si preveemos

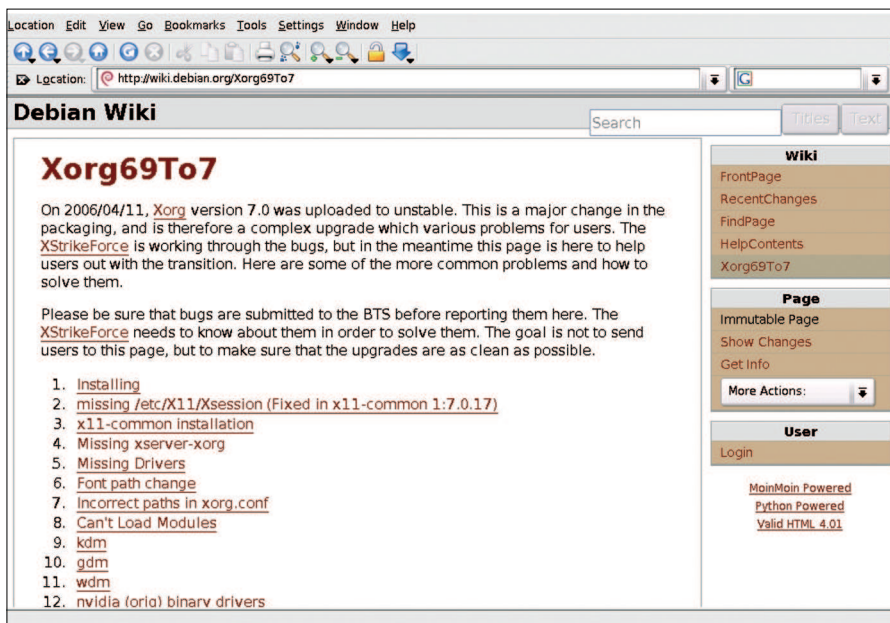


Figura 1: Cambiamos la configuración de la impresora con el botón derecho del ratón en el administrador de impresoras de KDE “kcmshell printmgr”.

que vamos a guardar sólo unos cuantos pero enormes archivos de imagen DMD o archivos de video sin comprimir, sería mejor elegir un número menor de inodos y un tamaño de bloque mayor si queremos desperdiciar menos espacio en disco.

Los sistemas de archivos del tipo btree + , como reiserfs o xfs usan un índice de archivos más o menos dinámico, que puede crecer o disminuir en función del número de archivos existente, y su estructura interna se parece más a un árbol que a una lista plana de archivos en secuencia continua. La ventaja de este diseño es, por supuesto, que prácticamente no hay limitaciones en cuanto al número de archivos guardado, y la manera en la que se balancea el árbol hace muy rápido el acceso a archivos. Por contra, es más complicado recuperar archivos en caso de que se dañe el índice por algún fallo en el disco duro, siendo más complicado el análisis minucioso de inconsistencias en la información. (Esto está relacionado con el hecho de que estos sistemas de archivos usan también logs de transacciones para verificar, repetir o restaurar el último estado consistente del sistema de archivos a la hora de montar la unidad).

NTFS y WinFS, al menos según los desarrolladores del linux-ntfs, parece que usan ambos métodos: el sistema de archivos en árbol balanceado más complicado que conocemos de reiserfs, mas el “viejo” índice de archivos estático que conocemos de DOS. Este último,

aparentemente añadido simplemente para que las operaciones de escritura parezcan más rápidas en lugar de los complicados métodos de inserción de archivos de un sistema de archivos basado en árbol balanceado. Nosotros usamos en estos momentos el método de escritura “tipo DOS” de NTFS en Knoppix, que está implementado en el proyecto libre de driver ntfs para Linux, por lo que verdaderamente PODEMOS escribir en particiones NTFS sin riesgo de dañar sus estructuras internas (y no del todo conocidas). El driver NTFS del kernel en las versiones 2.6.16/17 no tiene aún esta capacidad. Contaré más al respecto cuando vuelva a surgir la pregunta...

Configurar o Arreglar

? He estado siguiendo los últimos números de Linux Magazine y me he quedado maravillado al ver cómo le han contratado para escribir soluciones para algunos problemas con Linux. Lo que es más: me he quedado todavía más maravillado cuando pude leer su solución a la configuración de algunos CD y pude comprobar que sus recomendaciones eran prácticamente las mismas a las que yo había llegado siguiendo mi propia investigación.

Pero, ¿podría ayudarme con Gentoo? Me funcionan bastantes cosas, y acabo de optimizar Gnome hasta el punto de que lo podría utilizar realmente, si no fuera porque el sistema se vuelve no accesible. He aquí el escenario:

El arranque parece funcionar hasta el momento de loguearse. Veo una pantalla gráfica de login, tanto en KDE estándar como con Gnome.

Tecleo el nombre de usuario y la contraseña y pulso return. La ventana se vuelve del color elegido por defecto como fondo de la pantalla de login. El sistema muestra signos de arrancar un administrador de ventanas (durante unos dos segundos) y entonces aborta el proceso y vuelve a la pantalla de login, en lo que supongo que es una escapatoria del script.

Sólo hay una opción que funciona: loguearse como root en una consola. (Preguntaría cómo cambiarlo de manera que se permitiese ese login gráfico para el root, pero no creo que eso suponga un gran avance). Finalmente aparece una pista acerca de la causa: un mensaje de error que indica que ha fallado el login.

NO se trata de un nombre de usuario o contraseña erróneos. El nombre de usuario y su contraseña se aceptan claramente en el login. Se inicia *startkde* o *gnome*, pero finalizan tras dos o tres segundos.

Tiene pinta de que *login* está dañado o corrupto. O bien, algo va mal en las primeros pasos de X, KDE o Gnome.

El sistema funcionaba perfectamente al llamar a KDE o Gnome. Pero hace algunas semanas empezó a fallar de la manera que he descrito.

El único modo seguro que conozco de repararlo es reformatear la partición y empezar de nuevo. Pero eso no arregla nada realmente: sólo hace olvidar 3 ó 4 días de esfuerzos.

Comienzo a pensar que el problema puede estar relacionado con las X. No tengo claro si se completa el login con éxito. Tengo la sensación de que una parte del login está rota. Quizá se completa y el control de excepciones devuelve un error más tarde que provoca que el sistema vuelva a la pantalla de login. He probado reinstalando el paquete *pam_login* desde consola, pero no ha servido de nada.

Creo que la última vez que completé un login en modo gráfico fue con Gnome, que normalmente no utilizo. Podría decirse que el sistema está básicamente intacto, puesto que se puede llegar a una consola como root, pero en mi opinión el sistema falla a la hora de cargar el administrador de ventanas porque debe estar roto y necesito ser reparado. Si el sistema puede arreglarse reinstalando, ¿qué necesito reinstalar, sólo una parte o el sistema completo?



Este problema puede ser más fácil de arreglar de lo que piensa, sin que sea necesario reinstalar Gentoo.

Si hay un problema con *login*, o más bien, con los métodos de autenticación PAM, no llegaríamos ni siquiera al punto en el que la ventana de login (kdm, gdm o xdm) desaparece y comienza la sesión.

El procedimiento general en el login generalmente es:

1. El administrador del login se hace con tu contraseña o alguna clase de información de autenticación, bien a través del teclado, tarjeta inteligente, huella dactilar, o cualquier otro método que hayamos configurado en el sistema PAM.
2. Si se verifica la autenticación de manera correcta, el servidor X se alimenta con cierta configuración adicional (como tipos de letra, y autenticación local para programas X window), y
3. Se inicia la sesión X, usando nuestra ID de usuario.
4. Si finaliza la sesión X, el servidor X se resetea y podemos comenzar de nuevo en el paso 1.

Tras leer tu descripción, estoy seguro que ha llegado al paso 3. Xorg parece estar bien también, a menos que se pudiese identificar un punto donde el servidor X falla debido a un archivo fuente dañado o similar. Si has cambiado de Xorg 6.9 a 7.0 recientemente, deberías leer la documentación en <http://wiki.debian.org/Xorg69To7>.

Pero de momento, parece que el paso 3 termina de manera inesperada y se alcanza el paso 4.

Hay varios métodos para encontrar la razón por la que ocurre esto. La manera más fácil de depurar una sesión X es ir a la consola del sistema con Control-Alt-F1 y loguearse con nuestra cuenta de usuario. Podemos verificar el archivo *.xsession-errors* en nuestro directorio para encontrar por qué finalizó la última sesión.

Si estamos buscando una solución más exhaustiva:

1. Vamos a la consola del sistema con (en efecto) Control-Alt-F1, y nos logueamos como root (realmente).
2. Iniciamos otra instancia del servidor X sin especificar un archivo x-cookie:

```
X :1
```

La pantalla se volverá gris, y el servidor X se inicia en una nueva consola.

3. Volvemos atrás hasta la consola del sistema con Control-Alt-F1. Ahora, para conmutar a nuestra ID de usuario normal, tecleamos:

```
su - su_login_de_usuario
```

como root para obtener un shell de usuario. Si esto ya falla, hay algo verdaderamente mal en nuestra cuenta de usuario. Tal vez la ID de usuario ha cambiado por algún motivo, y ya no tenemos acceso a nuestro directorio de usuario. Deberíamos obtener el consecuente mensaje de error en este caso. Si por ejemplo se ha cambiado nuestro directorio de usuario a una ID de usuario diferente, debemos teclear esto como root para arreglarlo:

```
chown -R su_login_de_usuario \
/home/ysu_login_de_usuario
chmod -R u+rwX \
/home/su_login_de_usuario
```

(Cuidado con las opciones en mayúscula-minúscula). Pero supongamos que el directorio de usuario tiene los permisos correctos.

4. Abrimos un sencillo xterm en la pantalla X iniciada en el paso 2:

```
xterm -display :1
```

(Hacemos esto como usuario normal, no como root). Si preferimos un terminal gnome o konsole, debería funcionar también, si bien, puede que partes de Gnome o KDE se precargasen, y los resultados podrían variar de los obtenidos si intentamos loguearnos sólo con gdm/kdm.

5. Ahora volvemos a la consola donde se está ejecutando el servidor X :1. (Probamos con Control-Alt-F5 hasta F10: debería estar en alguno de éstos). Allí deberíamos encontrar una ventana xterm abierta donde podemos teclear comandos.
6. Intentamos iniciar nuestra sesión X a mano:

```
startkde
```

```
o
```

```
gnome-session
```

Nótese que cuando estos comandos nos

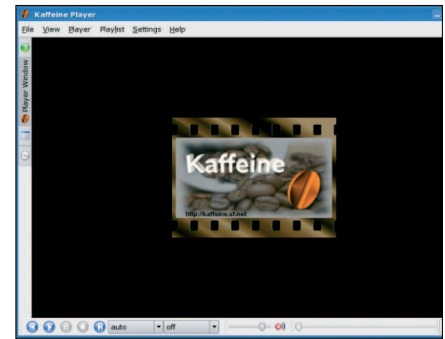


Figura 1: Cambiamos la configuración de la impresora con el botón derecho del ratón en el administrador de impresoras de KDE "kcmshell printmgr".

devuelven al prompt del shell, el servidor X finalizaría normalmente y veríamos de nuevo la ventana de login. No ocurre así en esta ocasión debido a que hemos iniciado el servidor de manera manual y no se le da la oportunidad a kdm o gdm a que lo controle.

Ahora deberíamos ser capaces de identificar el punto donde el sistema le está expulsando. Algunas veces se trata simplemente de una partición llena (i.e. KDE o GNOME no pueden crear los archivos temporales). Podría tratarse también de particiones erróneas en archivos esenciales del sistema, como */dev/null* o un directorio */tmp* donde no se puede escribir.

Una vez identificado y arreglado el error en el setup, podemos finalizar el servidor X en ejecución y volver a la pantalla de login para volverlo a intentar. Pero antes... deberíamos volver a probarlo con nuestro login kdm, por supuesto.

PCTV Card



Tengo un tarjeta PCTV Pinnacle 50e y me gustaría hacerla funcionar en mi equipo SUSE 10.0. ¿Cómo puedo configurarla? Si no fuese posible configurar esta tarjeta PCTV bajo Linux, ¿podría recomendarme alguna tarjeta PCTV que funcionase bajo Linux?



Desafortunadamente, no tengo esta tarjeta para probarla, pero hasta donde puedo comentarle, se supone que debería funcionar bajo un Linux con un kernel 2.6 reciente.

Para ver la TV y seleccionar canales, podría usar Kaffeine con el plugin DVB, que es parte de la distribución KDE 3.5.2 actual.

Puede encontrar esta información (y mucha más acerca de otras tarjetas de TV) en <http://www.bttv-gallery.de/>. ■