

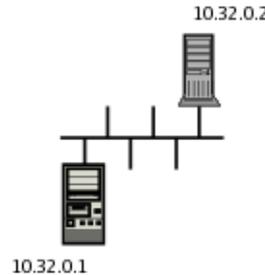


por Gerrit Renker
<gerrit.renker(at)gmx.de>

Sobre el autor:
Obtuvo su titulación en informática en el año 2.001.

Traducido al español por:
guadapeich ([homepage](#))

LF Tip: Clonando PCs completos por la red



Resumen:

Es frecuente encontrarse con la necesidad de replicar los datos desde un ordenador a otro. Un método seguro, sencillo y efectivo es utilizar un clonado por la red como se describe en este Cómo.

Escenario

Mientras que el clonado de animales ("la oveja Dolly") y de embriones humanos es un área de investigación muy discutida y dudosa, no hay duda que algún conocimiento en el clonado de ordenadores es menos peligroso (si está hecho correctamente) y más útil para quien escribe cuidadosamente scripts de configuración y opciones. Debido a la Ley de Moore y el rápido progreso en la fabricación de ordenadores, es casi seguro que uno se encontrará un escenario de replicado durante el uso de una distribución – puede ser que cambies el PC casero por un portátil o que un nuevo procesador más rápido sale a la venta y tienes bastante dinero para comprar un nuevo ordenador. La tarea es *tomar el sistema de archivos completo* del ordenador A y ponerlo a funcionar en el ordenador B. Hay dos maneras de hacerlo; una que no está explicada en este Cómo es abrir la carcasa, conectar físicamente los discos duros y hacer la copia localmente sobre un mismo ordenador. Esto frecuentemente no es posible – abrir la carcasa puede implicar la pérdida de la garantía – y también tiene riesgos ya que una persona inexperta puede dañar física y eléctricamente el equipo. Y más que eso, una vez perdí los datos de un disco de esta forma por culpa de un mal programa de bajo nivel. La otra aproximación, que requiere que ambos PCs tengan tarjetas de red (lo que es un caso frecuente en equipos caseros), es mucho más segura y se explica aquí.

La idea común subyacente a todos los métodos descritos a continuación es establecer una conexión de red entre el ordenador "*origen*" (el que va a ser clonado) y el ordenador "*destino*" (el clon). Esto es simple si ambos están conectados a un hub, en otro caso puedes conectar las tarjetas de red mediante un *cable cruzado* (los cables rectos normales no se pueden utilizar). Para el PC destino, un Live-CD (como Knoppix o LNX-BBC) o una instalación minimalista es necesaria de tal forma que la tarjeta de red esté operativa y `ssh` y/o `netcat` puedan ser usados. Hay también algunas distribuciones en disquete que te lo permitirán (en mi equipo `tomsrtb` no levantó la tarjeta de red al inicio). Si quieres instalar una nueva distribución, esa es

también una alternativa fácil. Ambos ordenadores necesitan ser configurados con direcciones IP en la misma red de forma que puedan "hablar" uno con otro, como se muestra en la ilustración superior.

Métodos posibles

Con la configuración básica descrita hay varias formas de realizar el clonado:

- copia binaria con `dd`
- tubería con `tar / cpio`
- `rsync`
- `dump` y `restore`

Lo primero es complicado, si no imposible, si tus discos duros no son del mismo tipo y con la misma geometría. Es grandioso para cosas como copiar imágenes iso (`dd if=/dev/cdrom of=the.iso`), o disquetes – [este \(script de copia de disquete\)](#) es un ejemplo de un script `diskcopy` utilizando `dd`. El otro inconveniente del método basado en `dd` es que el espacio vacío también será copiado, ocupando tiempo innecesariamente. La tubería de `tar` y `cpio` toma mucho tiempo (incluso horas) y tiene varios problemas. Por ejemplo, hay restricciones en el nombre de los ficheros y uso de los enlaces, perdiéndolos en los archivos en `/dev`, y similares. Yo no recomendaría esta solución para el replicado. Si tienes diferentes sistemas de ficheros en el origen y destino, **`rsync (1)`** es probablemente la mejor elección. Sólo necesita que `ssh` esté corriendo y transmitirá archivos eficazmente entre ambos mediante un eficiente protocolo incorporado. Incluso tiene una opción `-D` para los archivos de dispositivos, así como muchas otras opciones para utilizar en la mayoría de las situaciones posibles. Es una herramienta muy útil para las copias de seguridad diarias, copias espejo y cosas así, la página de manual con sus muchos ejemplos merece la pena estudiarla. Un ejemplo de clonado vía `rsync` se explica en [1].

Aquí, nosotros usamos el método mediante `dump` y `restore` el cual significa replicar el sistema de ficheros completo. Es rápido, efectivo y obtiene el resultado deseado con el mínimo esfuerzo, por lo tanto ideal para la replicación completa. Tuve que hacer realmente el procedimiento entero de clonado dos veces, puesto que el PC destino fue devuelto y sustituido. En ambos casos no se encontraron problemas y se obtuvo una copia arrancable funcional en el plazo de aproximadamente una hora de copiado de gigabytes. Este enfoque requiere que tanto el PC origen como el destino tengan el mismo tipo de sistema de ficheros. Asumimos que éste es `ext2` o `ext3`, puesto que son los tipos actualmente más usados (vea las notas [abajo](#)).

Configurando ssh

Una vez que hayas configurado un sistema con una instalación mínima o tengas en funcionamiento un Live-CD, el paso siguiente es instalar `ssh` (si no estas utilizando `netcat` para la transferencia según se explica más abajo). Esto requiere que el PC origen tenga `sshd` (the secure shell daemon) corriendo. Comprueba `/etc/init.d/` si dudas. En el PC destino, teclea como root:

```
ssh-keygen -t rsa
```

Para mantener las cosas simples, no entres una clave. La llave pública está en el archivo `/root/.ssh/id_rsa.pub`. Copia este archivo al PC origen mediante

```
scp /root/.ssh/id_rsa SourcePC:/tmp
```

donde `SourcePC` es la dirección IP del PC origen. Cuando pregunte si estas seguro, teclea un completo "yes" (solamente "y" algunas veces no funciona). Estas todavía siendo preguntado por la contraseña de root

en el PC origen. Ahora agrega el PC destino como un nodo de la red confiable mediante

```
cat /tmp/id_rsa.pub >> /root/.ssh/authorized_keys
```

en el PC origen. Para comprobar si todo está correcto, repite el anterior comando de copia en el PC destino. No serás preguntado mas por la contraseña.

Creando un sistema de ficheros en el PC destino

El primer paso es siempre particionar el disco duro en el ordenador destino y entonces crear el sistema de ficheros `ext2/ext3`. El último es activar la variante preferida de journaling simplemente colocando la opción `-j` (journaling) en `mke2fs` (requiere tener soporte `ext3` en el kernel). Puedes incluso convertir un sistema `ext2` a `ext3`, mira **tune2fs (8)**. Digamos que en el ordenador origen tenemos la configuración siguiente:

Partición	Tamaño	Usada	%Usada	Montada en
/dev/hda3	2.7G	552M	22%	/
/dev/hda5	7.8G	1.6G	22%	/usr
/dev/hda7	6.3G	1.7G	28%	/usr/share
/dev/hda8	3.4G	601M	19%	/home
/dev/hda12	5.3G	1.9G	37%	/opt
/dev/hda1	587M	70M	13%	/var/backup

Recomiendo hacer siempre alguna clase de partición. Si no en caso de fallo en el sistema de ficheros o corrupción de algunos sectores del disco duro se pueden destruir *todos* tus datos. Y de acuerdo a la Ley de Murphy, es seguro que sucederá si no tomas precauciones usando particiones diferentes en vez de usar una única. Tuve un caso recientemente con un kernel gracioso y no había particionado el disco, perdí todos mis datos sobre el sistema de ficheros corrupto. Arriba se muestra que `/usr` se hizo demasiado grande, por eso `/usr/share` ha sido añadido. Es el momento para conseguir un disco duro mayor.

En el PC destino lanza `parted` (recomendado) o tu particionador favorito (`Qtparted` es una bonita variante en modo gráfico, que dice ser un clon de Partition Magic). Crea particiones al menos tan grandes como las del PC origen. No olvides la partición swap. Después de salvar la tabla de particiones, instala un sistema de ficheros en todas la particiones recientemente creadas, usando

```
mke2fs -j -L <label> /dev/xxx
```

donde `xxx` es el nombre de la partición y `<label>` una etiqueta. Normalmente uso cosas como `"/usr"` como etiquetas (las verás en el arranque). Puedes configurar varias cosas mediante **tune2fs (8)**, como la revisión del sistema de ficheros de forma regular.

Transferir el sistema de ficheros

Primero necesitas montar todas las particiones nuevas. Comenzamos con la raíz del sistema (`/`) y montamos el resto de directorios. Es perfectamente posible unir dos particiones del PC origen en una sola en el destino, es lo que haremos con `/usr/` y `/usr/share` en el ejemplo anterior. Así monta tu futuro sistema raíz con

```
mount /dev/xxx /mnt
```

Al clonar es necesario cambiar al directorio destino

```
cd /mnt
```

Ahora la red, teclea en el PC destino

```
ssh targetPC 'dump -0 -f - /' | restore -r -f -
```

donde `targetPC` es la dirección IP del ordenador destino. Las opciones significan "-0" para una copia de seguridad completa, "-f -" le dice que use `stdin/stdout` como descriptor de archivo y "-r" indica a `restore` que recree el sistema de ficheros mediante volcado por la red en el PC destino. Para más opciones mira **dump (8)** y **restore (8)**. Abajo puedes ver la salida para transferir el sistema de ficheros raíz.

```
$ ssh 10.42.3.42 'dump -0 -f - /' | restore -r -f -
DUMP: Date of this level 0 dump: Tue Feb 22 15:50:12 2005
DUMP: Dumping /dev/hda3 (/) to standard output
DUMP: Label: debian
DUMP: Writing 10 Kilobyte records
DUMP: mapping (Pass I) [regular files]
DUMP: mapping (Pass II) [directories]
DUMP: estimated 547312 blocks.
DUMP: Volume 1 started with block 1 at: Tue Feb 22 15:50:14 2005
DUMP: dumping (Pass III) [directories]
DUMP: dumping (Pass IV) [regular files]
DUMP: Volume 1 completed at: Tue Feb 22 15:51:43 2005
DUMP: Volume 1 546590 blocks (533.78MB)
DUMP: Volume 1 took 0:01:29
DUMP: Volume 1 transfer rate: 6141 kB/s
DUMP: 546590 blocks (533.78MB)
DUMP: finished in 89 seconds, throughput 6141 kBytes/sec
DUMP: Date of this level 0 dump: Tue Feb 22 15:50:12 2005
DUMP: Date this dump completed: Tue Feb 22 15:51:43 2005
DUMP: Average transfer rate: 6141 kB/s
DUMP: DUMP IS DONE
```

`Restore` siempre crea un archivo `restoresymtable` el cual puede ser borrado una vez que estas convencido de que no se han producido errores durante la reconstrucción del sistema de ficheros. Finalizado el sistema de ficheros raíz, ahora procedemos con cada partición, comenzando con `/usr` (asumiendo que tu actual directorio de trabajo es la raíz del futuro sistema de ficheros).

```
mount /dev/xxx ./usr
```

```
cd ./usr
```

```
ssh targetPC 'dump -0 -f - /usr' | restore -r -f -
```

El ciclo `mount-cd-dump/restore` se repite para todos los directorios que tengas. Con miras a `/usr/share` (el cual en el PC origen tiene su propia partición) puedes, después del paso anterior, simplemente cambiar al directorio `./usr/share` (fíjate en el ".") y entonces repetir

```
ssh targetPC 'dump -0 -f - /usr/share' | restore -r -f -
```

`Restore` solo se queja si los archivos existen en el sistema de ficheros que está siendo restaurado, así cuando ponemos dos particiones diferentes del PC origen en el PC destino no hay problema. Clonar un PC completo tomó sobre una hora con `ssh` y tarjetas de red de 100MB (con cable cruzado algo más).

Nota: Para volcar un sistema de ficheros, no es necesario que esté montado. Puedes también pasar un *nombre de partición*, como `/dev/hda6`, en vez del nombre del directorio donde se monta la partición.

Alternativa: netcat

Una alternativa, en vez de usar `ssh`, es `netcat` (1) que se abrevia como `nc`. Netcat es la navaja suiza de los TCP/IP cliente-servidor y permite crear una tubería sobre la red. El ejemplo anterior es entonces modificado como sigue. Asumimos que la partición montada bajo `/var/backup` va a ser transferida mediante `dump/restore` desde el PC origen a el PC destino.

En el punto de llegada (*PC destino*) crea una sesión de escucha de `netcat` vía `-l` que vuelque su salida a `restore`.

```
nc -l -p 2000 -q 1 | restore -r -f -
```

En el *PC origen* crea otra sesión de `netcat` que tome su entrada desde una tubería donde `target-IP` es la dirección IP del PC destino.

```
dump -0 -f - /var/backup | nc <target-ip> 2000
```

La opción `-q` es para parar `nc` después de recibir el final del fichero, pero yo tengo que terminarlo manualmente en mi caso. Recomiendo usar `ssh` de todas formas.

Limpiando

Felicidades, si lo conseguiste básicamente has clonado un sistema. Ahora queda convertir esta copia en un PC usable. Lo primero es actualizar `/etc/fstab` con los nuevos datos, o no serás capaz de usar las particiones clonadas. Si la dirección IP cambia, también necesitas actualizarla (`/etc/hosts`, `/etc/network/interfaces` en `debian`). La siguiente cosa importante es la *configuración de arranque* que en la mayoría de los casos debe ser actualizada. Con `lilo`, edita `/etc/lilo.conf` (en particular la opción `root=...`) y entonces corre `lilo -v`. Con `grub`, edita `/boot/grub/menu.lst` (o `/boot/grub/grub.conf` dependiendo sobre cual esté el enlace) y entonces teclea `grub`,

```
grub> root (hd0,xxx)
... filesystem is ...
grub> setup (hd0)
... lots of output here
grub> quit
```

o corre `grub-install /dev/xxx` donde `xxx` es tu disco duro. Aquí comprueba ambos, `root (hdn,xxx)` y la configuración añadida a `root=/dev/xxx`.

En el caso común de que ahora tengas mejor hardware en el PC clonado nuevo, querrás actualizar la *configuración de tu kernel a medida*. Si estas usando un sistema que viene con montones de módulos preconfigurados (como RedHat, SuSe, Mandrake, Fedora ...) no te preocupes, es bastante posible que exista un módulo utilizable. En caso contrario, `lspci -vv` y compilar el kernel es algo común y está descrito en muchos sitios. Si tu tarjeta gráfica es diferente ahora, actualiza `/etc/X11/XF86Config-4` (o `xorg.conf` en RH/Fedora) para reflejarlo, o no tendrás salida gráfica. Es posible usar las herramientas de gráficos para configurar X arrancando en el nivel 3 si tienes esas herramientas. En `debian`, alguna investigación es necesaria, yo fui afortunado al encontrar que el controlador cambió de `r128` a `radeon` y eso fue lo que hice.

Otros sistemas

Este cómo explica el procedimiento de clonado para sistemas de ficheros `ext2/ext3`. Órdenes muy similares se pueden encontrar en muchos otros sistemas. Por ejemplo, varios Unices como FreeBSD, HP-UX, IRIX también suministran las órdenes `dump/restore`; en Solaris se llama `ufsdump/ufsrestore`. Hay sistemas de ficheros que no ofrecen la utilidad de volcado, por ejemplo ReiserFS. En este caso recomendaría usar `rsync`. Mira [1] para un informe de clonado de un sistema Linux con éxito.

Referencias

[1] "*Replicating a Linux System – Yet Another Method.*" Ben Okopnik, Linux Gazette Issue 83, October 2002.

<p><u>Contactar con el equipo de LinuFocus</u> © Gerrit Renker "some rights reserved" see linuxfocus.org/license/ http://www.LinuxFocus.org</p>	<p>Información sobre la traducción: en --> --- : Gerrit Renker <gerrit.renker(at)gmx.de> en --> es: guadapeich (homepage)</p>
--	---

2005-05-19, generated by lfparsr_pdf version 2.51