

Un pingüino en el bolsillo

coLinux permite que funcione la ejecución de dos sistemas operativos

Poder utilizar aplicaciones de otros sistemas operativos desde tu propio entorno cada vez es más sencillo, ahora incluso sin herramientas de virtualización. Veamos cómo hacerlo.

VMWare y Virtual PC son dos de las aplicaciones más conocidas para este tipo de situaciones. Ambas son capaces de crear una nueva máquina virtual en nuestro ordenador, haciendo que sea posible ejecutar múltiples sistemas operativos de forma simultánea. Se trata, por supuesto, de aplicaciones comerciales, que disponen de un número de opciones elevado y cuyo precio todavía es algo excesivo (en torno a los 200 \$). Hay, sin embargo, otras opciones que pueden ser útiles en casos donde el presupuesto es nulo. Bochs, por ejemplo, es un proyecto Open Source capaz de emular un ordenador completo basado en la arquitectura x86. Esto quiere decir que podemos ejecutar un completo sistema operativo MS-DOS o similar.

Nos parece mucho más interesante otro proyecto, uno con el cual podemos ejecutar una distribución de Linux casi sin modificar. Al contrario de lo que hace Bochs, coLinux (así es como se llama este proyecto) no emula el hardware, pero hace que conviva la ejecución de los dos sistemas operativos. Esta diferencia fundamental se traduce en un rendimiento muy superior por parte de coLinux.

Pero, como todo, también tiene algunos

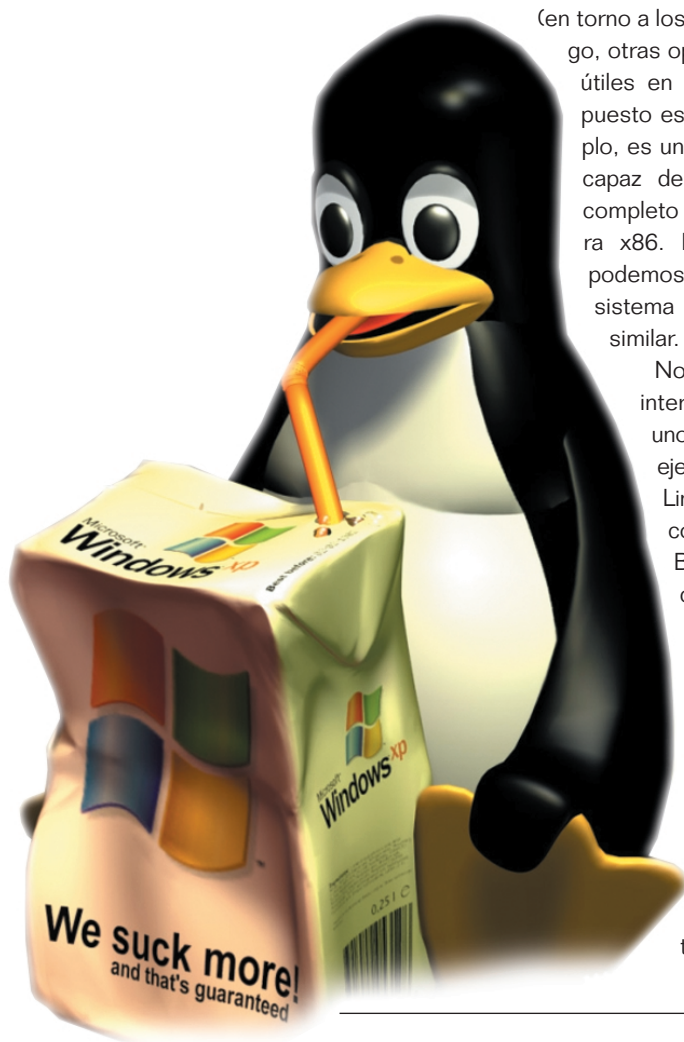
inconvenientes. El más grande es la dificultad de su instalación. Mientras que con VMWare Workstation podemos tener una distribución de Linux instalada en menos de media hora, con las herramientas proporcionadas en este proyecto experimental tardaremos algo más. De hecho, conviene tener algo de soltura con esta plataforma Open Source para instalarlo correctamente.

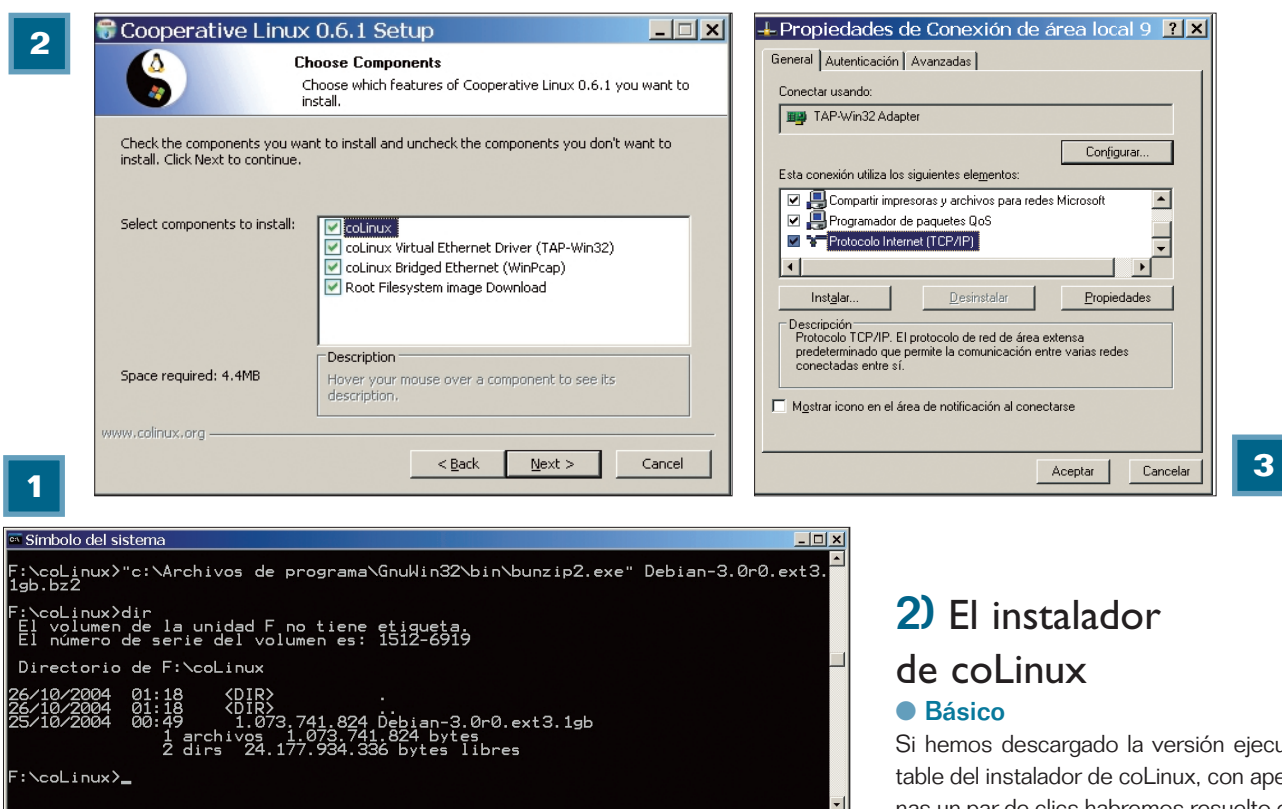
Pre-requisitos

Antes de comenzar, conviene tener algunas herramientas disponibles. Sin duda, la más importante es el propio coLinux. (<http://www.colinux.org/?section=downloads>) y el archivo raíz necesario para su ejecución. Éste no es más que una distribución de Linux previamente instalada y muy «limpia», esperando a que nosotros la personalizemos. Podemos elegir entre varias, desde Fedora Core hasta Debian. Los que prefieran compilar todo aquello que tocan encontrarán una valiosa Gentoo.

En nuestro caso, escogimos una Debian basada en el kernel 2.4, por propia familiaridad con esta distribución y tratarse de la primera que se adaptó a coLinux. El kernel también puede escogerse, pero cuidado: para que el núcleo de Linux conviva con el de Windows es necesario «parchear» el primero (e instalarle un controlador en el segundo), por lo que «cualquier» kernel no vale (como tampoco funcionará cualquier distribución). Para no complicar demasiado la instalación utilizaremos un kernel ya parcheado de la rama 2.4.

Esto quiere decir que deberíamos reunir al menos tres ficheros: «Debian-3.0r0.ext3.1gb.bz2» (la distribución) y «coLinux-0.6.1.exe» (coLinux). Puede que





el número de versión cambie, en especial el del propio proyecto, por lo que tened cuidado a la hora de escoger una revisión más reciente: el proceso podría no ser el mismo.

Además conviene descargar alguna que otra herramienta adicional, como por ejemplo Putty, un cliente SSH para Windows que podéis encontrar en <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>, o un editor de texto con soporte para ficheros ASCII de tipo Unix., como Crimson (<http://www.crimsoneditor.com/>, que hace esto y mucho más).

Preparad también algo más de 1,5 Gbytes de disco duro en una partición para instalar el sistema raíz de Linux. Es posible utilizar una partición al completo. Sin embargo, vamos a procurar mantener las cosas lo más simple posible.

Vamos a obviar, por ahora, los pasos necesarios para hacer funcionar XWindows, pero aquellos que necesitan de un entorno gráfico pueden recurrir al Wiki del proyecto para buscar más información: <http://www.colinux.org/wiki/index.php>.

1) La partición

● Básico

Antes de comenzar con coLinux hemos de prepararle un poco el entorno en el que se va a instalar. Normalmente, el instalador de la aplicación se encargará de todo, incluso de descargar el sistema de ficheros raíz. Pero como lo hemos descargado previamente necesitaremos descomprimirlo del formato en el que se encuentra, en este caso bzip2. Este tipo de archivos son comparables a los «.zip» que maneja Windows, pero utiliza otro tipo de aplicación de compresión/descompresión, por lo que vamos a necesitarla. Encontraréis una herramienta Open Source para descomprimir el fichero en <http://gnuwin32.sourceforge.net/packages/bzip2.htm> y descargar el archivo *Setup* llamado *Complete package, except sources*. Instaladlo en su lugar por defecto (que será algo como *c:\Archivos de Programa\GnuWin32* y descomprimid el archivo (desde el directorio en el que se encuentra y en la línea de comando "*c:\Archivos de programa\GnuWin32\bin\bunzip2.exe Debian-3.0r0.ext3.1gb.bz2*". Tras unos segundos tendremos preparada la partición.

2) El instalador de coLinux

● Básico

Si hemos descargado la versión ejecutable del instalador de coLinux, con apenas un par de clics habremos resuelto el asunto. En primer lugar, tendremos que escoger qué tipo de instalación queremos realizar. Obviamente, hemos de seleccionar *coLinux*. También deberemos indicar el tipo de red que queremos utilizar, en forma de un adaptador nuevo para Windows (virtual), o utilizar uno ya existente con WinPcap. De momento, os recomendamos decantaros por el uso de un adaptador virtual. Éste «combina» mejor con el resto del sistema y en múltiples configuraciones ya que, por ejemplo, WinPcap puede tener problemas con algunos adaptadores de red. Podéis permitir con tranquilidad la instalación del controlador virtual.

3) La configuración de red Windows

● ● Intermedio

Para que el sistema operativo «hospedado» pueda utilizar la tarjeta de red de Windows han ideado un curioso sistema. Utilizando un adaptador virtual (el que hemos instalado durante el asistente), «conectamos» Windows a una nueva tarjeta de red. coLinux es capaz de comunicarse con esta tarjeta de red a través de un «cable virtual cruzado».

4

```

F:\coLinux>"c:\Archivos de programa\GnuWin32\bin\bunzip2.exe" swap_1024Mb.bz2
F:\coLinux>dir
El volumen de la unidad F no tiene etiqueta.
El número de serie del volumen es: 1512-6919

Directorio de F:\coLinux
26/10/2004 01:50 <DIR> .
26/10/2004 01:50 <DIR> ..
25/10/2004 00:49 1.073.741.824 Debian-3.0r0.ext3.1gb
26/10/2004 01:46 1.073.741.824 swap_1024Mb
                2 archivos 2.147.483.648 bytes
                2 dirs 23.104.196.608 bytes libres
F:\coLinux>

```

Linux necesitará una partición virtual; será necesario que la creamos descargando un fichero de un tamaño parecido a la memoria del PC.

En realidad, el adaptador utiliza las API de TAP-32 para comunicar ambos sistemas, huésped y anfitrión. Pero esta tarjeta ha de tener una dirección IP, para lo cual abriremos las *Conexiones de Red* y buscaremos la conexión de *TAP-Win32*. Ésta muestra el cable desconectado, lo que quiere decir que coLinux todavía no está «encendido». Pulsad sobre sus *Propiedades* y seleccionad las propiedades del *Protocolo TCP/IP*. En esta nueva ventana dotaremos a la interfaz con una nueva dirección IP y máscara de subred. En nuestro caso utilizamos la dirección privada 192.168.0.1 en la subred 255.255.255.0.

4) La partición de swap

● Básico

Al igual que Windows requiere un fichero de intercambio, Linux necesitará una partición (virtual) de *swap*. Para crearla tan sólo hay que descar-

gar el fichero del tamaño que más se ajuste a la memoria de vuestro PC. La regla que seguir es la de igualar la memoria real con la virtual. Nosotros teníamos 1 Gbyte y por lo tanto descargamos *swap_1024Mb.bz2*. Al descomprimirlo, como hicimos con la partición raíz, obtendremos un fichero «gigante» que coLinux utilizará tras configurarlo.

5) El fichero de configuración

● ● Intermedio

El archivo que controla el comportamiento de coLinux en nuestra máquina es *default.colinux.xml* y se encuentra en la misma carpeta en la que instalamos esta aplicación. Para editarla podemos utilizar en realidad cualquier editor de texto (incluso el *Bloc de Notas*) y los parámetros que hemos de modificar son bastante limitados. Las líneas marcadas con

La cantidad de memoria que tenemos que reservar a este mágico coLinux ha de ser de tan sólo 128 megabytes de nuestro sistema operativo

5

```

<!-- This line needs to point to your root file system.
For example change "root_fs" to the name of the Debian image.
Inside colinux it will be /dev/cobd0 -->
<block_device index="0" path=""\bosDevices\c:\coLinux\root_fs" enabled="true"/>
<!--
This line can specify a swap file if you wish, or an additional
image file, it will /dev/cobd1. Additional block devices can
be specified in the same manner by increasing the index
-->
<block_device index="1" path=""\bosDevices\c:\coLinux\swap_device" enabled="true"/>
<!--
bootparams allows you to pass kernel boot parameters
-->
<bootparams>root=/dev/cobd0/bootparams
<!-- image allows you to specify the kernel to boot -->
<image path=""vmlinuz"/>
<!--
this line allows you to specify the amount of memory available
to colinux
-->
<memory size="64"/>
<!--
This allows you to modify networking parameters, see the README
or website for more information
-->
<network index="0" type="tcp"/>
</colinux>

```

block_device reflejan las diferentes particiones con las que contará la máquina virtual. En nuestro caso, hemos de modificar la ruta de los archivos que hacen de partición para que se ajuste a nuestro sistema. Existen dos de estas líneas, las cuales deben reflejar las dos particiones que hemos preparado previamente, la *root* y el *swap*.

Tras realizar estos dos cambios aclararemos que podemos utilizar múltiples interfaces de red virtuales para múltiples máquinas, por eso se encuentra aquí el parámetro *Network*, para diferenciar entre ellos.

Un último parámetro importante es el de la memoria reservada para coLinux. A la hora de ajustarlo, tened en cuenta que esta memoria queda

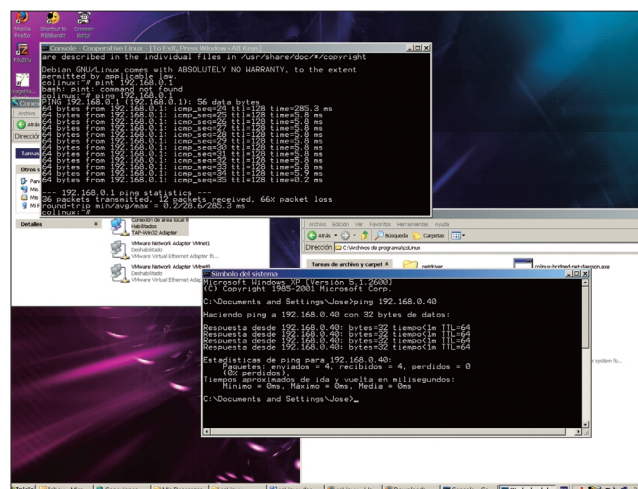
```

modprobe: Can't open dependencies file /lib/modules/2.4.26-co-0.6.1/modules.dep
(No such file or directory)
Checking all file systems...
fsck 1.27 (8-Mar-2002)
Setting kernel variables.
Mounting local filesystems...
nothing was mounted
Running Odns-down to make sure resolv.conf is ok...done.
Cleaning: /etc/network/ifstate.
Setting up IP spoofing protection: rp_filter.
Configuring network interfaces: Ignoring unknown interface lo=lo.
done.
Cleaning: /tmp /var/lock /var/run.
Initializing random number generator... done.
Recovering nvi editor sessions... done.
INIT: Entering runlevel: 2
Starting system log daemon: syslogd.
Starting kernel log daemon: klogd.
Starting internet superserver: inetd.
Starting deferred execution scheduler: atd.
Starting periodic command scheduler: cron.
Debian GNU/Linux 3.0 colinux tty1
colinux login:

```

6

Iniciamos coLinux por primera vez y comprobamos lo familiar que resulta el entorno.



Es preciso activar la función para compartir la conexión a Internet a través de las Propiedades Avanzadas.

reservada exclusivamente para este sistema operativo y por lo tanto no puede ser usada por Windows. Si elevamos demasiado esta cantidad, puede que tengamos algún que otro problema de estabilidad. En nuestro caso, dejamos el valor en 128 Mbytes.

6) La primera vez

● Básico

Para iniciar coLinux por primera vez tan sólo hemos de dirigirnos a la carpeta en la que lo instalamos desde la línea de comando. Desde ella hemos de ejecutar «colinux-daemon.exe -t nt -c default.colinux.xml», ante lo cual la ventana comenzará con un desfile de líneas de texto. En él se puede ver cómo arrancan el kernel de Linux y todos sus subsistemas normalmente. Hasta hace poco la consola de coLinux era bastante limitada. Sin embargo, el avance del proyecto ha conseguido algo mucho más familiar con el entorno «nativo». Como veréis, se trata de una adaptación de la consola convencional de Windows, por lo que los controles para el ajuste del tipo, tamaño y color son personalizables, eso sin contar con que es operativa la función de cortar y pegar.

Pero mientras el nuevo sistema operativo se iniciaba, también lo ha hecho el sistema de red, y si nos dirigimos a la ventana de *Conexiones de red* veremos que ahora el cable sí está conectado.

El sistema de red se inicia al mismo tiempo que el nuevo sistema operativo

7) Conexión entre los dos equipos

● Intermedio

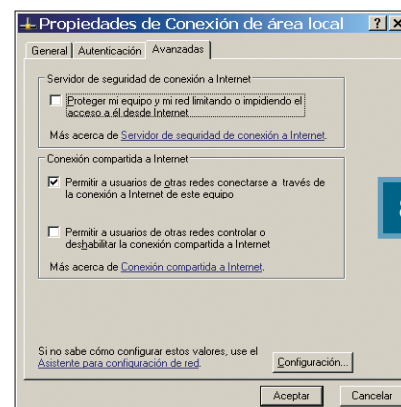
Aunque el cable virtual comunica ambos sistemas operativos, a Linux no le hemos asignado todavía ninguna dirección IP. En teoría, este paso es imprescindible, pero la distribución que hemos descargado ya se ha configurado con la dirección: 192.168.0.40, que «casualmente» se encuentra dentro de la misma subred que el otro lado del cable, el adaptador de red virtual. La ruta tampoco la hemos de configurar, como comprobamos tras teclear «route» en la consola. Sobra decir que el usuario de entrada será *root* sin contraseña. Si hacemos ahora *ping 192.168.0.1*, veremos que «Windows» responde, ocurriendo lo mismo si repetimos el proceso desde el otro sistema operativo. Pero esto no quiere decir que tengamos conexión a Internet. Por dos motivos: porque no hemos configurado los servidores DNS (que conseguiremos editan-

do el fichero «/etc/resolv.conf» con vi) y porque Windows no está configurado para encaminar los paquetes que llegan desde esta tarjeta.

8) Windows como encaminador

● Intermedio

Para conseguir que los paquetes de coLinux salgan al exterior, basta en realidad con activar la función para compartir la conexión a Internet del que se encuentra en la pestaña *Avanzadas* de las propiedades del adaptador con el que nos conectamos a Internet. Un ping a *google.com* o un *apt-get update* debe-



ría mostrar ahora resultados satisfactorios. Comprobad que habéis establecido correctamente las direcciones IP y que no habéis omitido la configuración de los servidores DNS si algo falla. Probablemente os interese configurar algunos paquetes como los de teclado.