LILO: Linux Loader

Un estándar de facto

Pablo Garaizar Sagarminaga
txipinet@txipinet.com

LILO se ha ganado un puesto como estándar entre los gestores de arranque. En este artículo conoceremos sus características más reseñables y aprenderemos a configurarlo para exprimir su potencia y protegernos ante posibles intrusos.

El proceso de arranque del Sistema Operativo

Quizá mucha gente no se ha parado nunca a pensar cómo nuestros PCs son capaces de saber qué ficheros hay que ejecutar para cargar correctamente el Sistema Operativo (SO en lo sucesivo) de manera automática. En un principio parece obvio, encendemos el ordenador, y se carga el sistema sin más. Todos vosotros sabéis que esto no es tan sencillo. La gran mayoría hemos tenido que particionar nuestros discos duros para instalar la partición raíz del sistema, instalar un gestor de arranque del SO, configurarlo, etc. Tal vez, si fuimos muy rápido con la instalación, nos quedamos sin poder acceder a otros sistemas que no fueran Linux, tal vez todavía tengamos que arrancar desde diskette si queremos gozar de nuestro Linux, o tal vez todo funcionó correctamente... todo ello lo entenderíamos mejor teniendo claro el proceso de arranque del SO.

Nada más pulsar el botón de encendido de nuestro equipo, se alimenta la placa base del ordenador. Después de unos chequeos el sistema (POST: Power On Self Test) pasa el control a la memoria BIOS (Basic Input / Output System). Hoy en día todas las BIOS son configurables por lo que en este momento la BIOS buscará un sistema en los lugares que nosotros le hayamos indicado. Una configuración típica sería buscar primero en la disquetera (floppy), posteriormente en la unidad de CD-ROM y por último en el primer disco IDE (IDE-0). Esta configuración es la responsable, por ejemplo, de los típicos mensajes de error cuando nos olvidamos un diskette dentro de la unidad. Si bien esto es lo más usual, en determinadas circunstancias puede suponer una debilidad en cuanto a la seguridad del sistema.

De esta manera estamos permitiendo que alguien con un SO en diskettes (Trinux, por ejemplo) pueda arrancar su sistema y montar posteriormente nuestro disco duro, accediendo a todos los datos que puediera contener. Por este motivo es conveniente configurar nuestros sistemas para que no busquen un SO en dispositivos externos y proteger nuestra BIOS mediante una clave. Éste no es un método infalible, pero dificulta bastante las cosas a posibles intrusos.

Figura 1. Configuración típica de la secuencia de arranque en la BIOS.

Una vez establecida la unidad desde la que se arrancará el SO, se carga el sector de arranque (Boot Sector), que contiene un pequeño programa (inferior al tamaño de 1 sector, 512 bytes) que indica qué es lo siguiente que hay que leer para continuar con la carga del SO. En SOs primitivos como MS-DOS este programa consistía casi únicamente en un salto al sector de disco que contenía el núcleo del SO. Actualmente los sectores de arranque suelen consistir en un pequeño menú que indica las posibles opciones en cuanto a SOs disponibles. Así tenemos el "NTLDR" o el "Boot Manager" de parte del software propietario o bien el "LILO" o el "GRUB" de la mano del software libre. Estos "Boot Loaders" contienen los apuntadores a los diferentes sectores en disco que permiten arrancar cada uno de los SOs instalados en nuestros sistemas.

Así pues, resumiendo, el proceso sería el siguiente:

- encendemos la máquina
- se realiza un autochequeo (POTS)
- se atiende a la configuración de la secuencia de arranque (Boot Sequence)
- se busca en el primer sector del disco eledgido el "Boot Loader"
- en función de las características de ese "loader" o cargador, se elige uno u otro SO
- se procede a la carga del SO desde disco

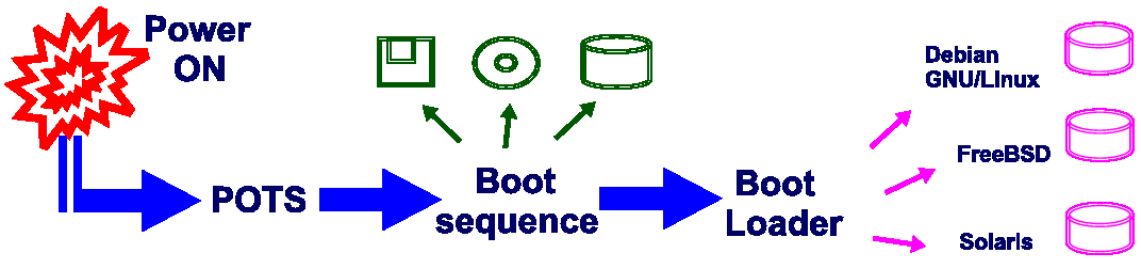

Figura 2. Proceso de arranque del Sistema Operativo en un PC.

Gestores de arranque en Linux

Existen varios "Boot Loaders" en Linux, muchos de ellos sirven para arrancar varios tipos diferentes de SOs y otros sirver para poder arrancar Linux desde otras plataformas. Hoy en día tienen relevancia sólo unos pocos:

- LILO: LInux LOader, el más conocido y usado.
- GRUB: GRand Unified Bootloader, un "Boot Loader" muy potente y flexible que está ganando terreno.
- Loadlin: muy utilizado para poder cargar Linux desde DOS. Incluído en la práctica totalidad de las distribuciones.
- SYSLinux: "Boot Loader" rápido y eficiente, utilizado mayoritariamente por distribuciones de diskette o CDs de instalación.
- Poof: permite arrancar Linux desde iMac’s, facilitando la expansión del software libre en entornos Mac.

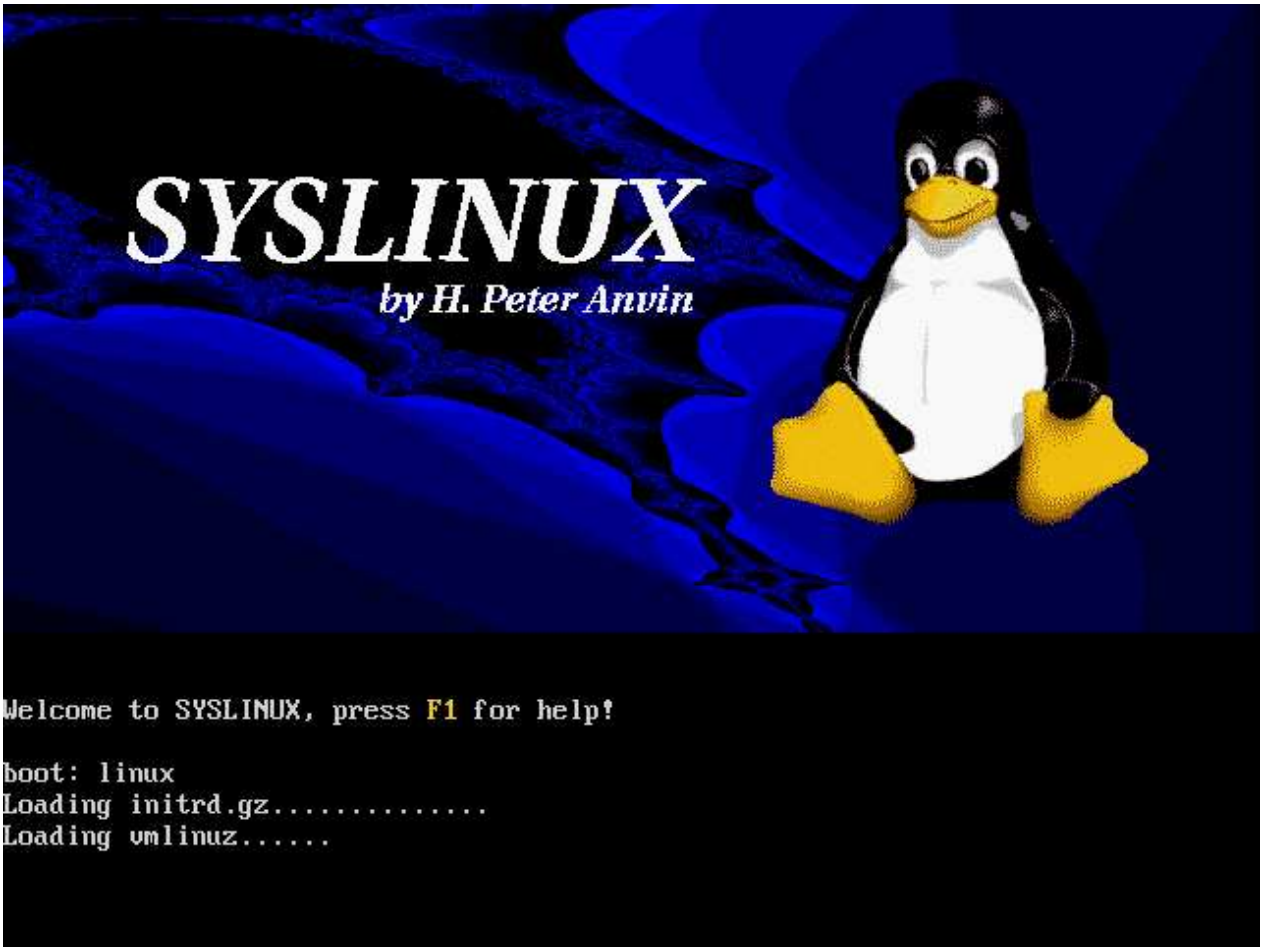

Figura 3. SYSLinux, un "Boot Loader" muy completo y eficiente.

Todos ellos tienen su espacio en el panorama actual, si bien LILO es el gran dominador en el cómputo global.

LILO, LInux LOader

Ya hemos visto como funciona el proceso de arranque y supongo que entenderemos que cuando LILO arranca el sistema, utiliza llamadas a la BIOS para volcar el kernel de Linux de disco a memoria. De esto se deduce que el kernel del sistema deberá residir en un dispositivo accesible por la BIOS (nuestra partición raíz podrá, sin embargo, residir en un dispositivo no accesible por la BIOS, como un disco duro SCSI).

Un hecho llamativo del uso de LILO es que deberemos reinstalarlo cada vez que varíe nuestra configuración de arranque. Suele ser muy habitual modificar el fichero de configuración de LILO (/etc/lilo.conf) y no actualizar el sector de arranque ejecutando "lilo" (/sbin/lilo). El motivo de tener que reinstalarlo cada vez es que LILO no es capaz de acceder a los sistemas de ficheros en tiempo de arranque, no sabe montar discos, entonces lo que hace es convertir los datos de su fichero de configuración (/etc/lilo.conf) en direcciones absolutas dentro del disco (similar al proceso de compilación de un

programa con direcciones de memoria absolutas).

A la hora de instalar LILO la pregunta típica es ¿dónde desea instalar LILO, en el MBR o en la partición raíz del sistema? La respuesta a esta pregunta está en función del resto de SOs que tengamos instalados en nuestra máquina. En líneas generales, si disponemos de otro SO con otro "Boot Loader" y queremos conservarlo, deberemos instalar LILO en la partición raíz (/dev/hda2, por ejemplo). Si sólo vamos a tener Linux en nuestro equipo o si queremos que el arranque se gestione a través de LILO, deberemos elegir instalarlo en el MBR (esto no implica que otro SO no pueda ser accesible desde LILO, más adelante veremos cómo hacerlo).

En los siguientes apartados aprenderemos a configurar LILO y a entender su fichero de configuración (/etc/lilo.conf).

Convivencia con otros Sistemas Operativos

El escenario típico de configuración de LILO es una estación de trabajo que requiere un arranque dual, esto es, poder arrancar con Linux y con otro SO a petición del usuario. Vamos a ver un ejemplo típico de configuración de LILO y entenderemos cómo es posible hacer esto:

```
argon:~# cat /etc/lilo.conf
# /etc/lilo.conf - See: 'lilo(8)' and 'lilo.conf(5)',
# ---------------       'install-mbr(8)', '/usr/share/doc/lilo/',
#                       and '/usr/share/doc/mbr/'.

# +---------------------------------------------------------------+
# |                         !! Reminder !!                        |
# |                                                               |
# | Don't forget to run 'lilo' after you make changes to this     |
# | conffile, '/boot/bootmess.txt', or install a new kernel.  The |
# | computer will most likely fail to boot if a kernel-image      |
# | post-install script or you don't remember to run 'lilo'.      |
# |                                                               |
# +---------------------------------------------------------------+

# Support LBA for large hard disks.
#
lba32

# Specifies the boot device.  This is where Lilo installs its boot
# block.  It can be either a partition, or the raw device, in which
# case it installs in the MBR, and will overwrite the current MBR.
#
boot=/dev/hda

# Specifies the device that should be mounted as root. ('/')
#
root=/dev/hda2

# Enable map compaction:
# Tries to merge read requests for adjacent sectors into a single
# read request. This drastically reduces load time and keeps the
# map smaller.  Using 'compact' is especially recommended when
# booting from a floppy disk.  It is disabled here by default
```

```
# because it doesn't always work.
#
# compact

# Installs the specified file as the new boot sector
#
install=/boot/boot.b

# Specifies the location of the map file
#
map=/boot/map

# You can set a password here, and uncomment the 'restricted' lines
# in the image definitions below to make it so that a password must
# be typed to boot anything but a default configuration.  If a
# command line is given, other than one specified by an 'append'
# statement in 'lilo.conf', the password will be required, but a
# standard default boot will not require one.
#
# This will, for instance, prevent anyone with access to the
# console from booting with something like 'Linux init=/bin/sh',
# and thus becoming 'root' without proper authorization.
#
# Note that if you really need this type of security, you will
# likely also want to use 'install-mbr' to reconfigure the MBR
# program, as well as set up your BIOS to disallow booting from
# removable disk or CD-ROM, then put a password on getting into the
# BIOS configuration as well.  Please RTFM 'install-mbr(8)'.
#
# password=tatercounter2000

# Specifies the number of deciseconds (0.1 seconds) LILO should
# wait before booting the first image.
#
delay=20

# You can put a customized boot message up if you like.  If you use
# 'prompt', and this computer may need to reboot unattended, you
# must specify a 'timeout', or it will sit there forever waiting
# for a keypress.  'single-key' goes with the 'alias' lines in the
# 'image' configurations below.  eg: You can press '1' to boot
# 'Linux', '2' to boot 'LinuxOLD', if you uncomment the 'alias'.
#
# message=/boot/bootmess.txt
    prompt
#       single-key
#       delay=100
        timeout=100

# Specifies the VGA text mode at boot time. (normal, extended, ask, )
#
# vga=9
#
vga=normal

# Kernel command line options that apply to all installed images go
# here.  See: The 'boot-prompt-HOWO' and 'kernel-parameters.txt' in
```

```
# the Linux kernel 'Documentation' directory.
#
# append=""

# Boot up Linux by default.
#
default=Linux

image=/vmlinuz
        label=Linux
        read-only
#       restricted
#       alias=1

image=/vmlinuz.old
        label=LinuxOLD
        read-only
        optional
#       restricted
#       alias=2

# If you have another OS on this machine to boot, you can uncomment the
# following lines, changing the device name on the 'other' line to
# where your other OS' partition is.
#
other=/dev/hda1
        label=FreeBSD
#       restricted
#       alias=3
```

El fichero de configuración (/etc/lilo.conf) es bastante autoexplicativo, pero comentaremos algunos puntos interesantes:

- La cláusula "boot=" especifica dónde se instalará LILO (MBR o partición raíz), en este caso se ha instalado en el MBR ("boot=/dev/hda").
- La cláusula "root=" indica cuál será la partición raíz del sistema ("root=/dev/hda2")
- Con "install=/boot/boot.b" indicamos que ese fichero contiene el sector de arranque que queremos instalar. Normalmente se trata de un enlace simbólico a cada una de las posibilidades que nos ofrece LILO:

```
argon:/boot# ls -al
total 1376
drwxr-xr-x    2 root     root         4096 Jan 17 07:27 .
drwxr-xr-x   19 root     root         4096 Nov 21 09:19 ..
-rw-r--r--    1 root     root       265635 Nov 21 09:17 System.map-2.2.17
-rw-r--r--    1 root     root         7708 Dec 17 07:53 boot-bmp.b
-rw-r--r--    1 root     root         5920 Dec 17 07:53 boot-compat.b
-rw-r--r--    1 root     root         7680 Dec 17 07:53 boot-menu.b
-rw-r--r--    1 root     root         5920 Dec 17 07:53 boot-text.b
-rw-r--r--    1 root     root          512 Nov 21 09:19 boot.0300
lrwxrwxrwx    1 root     root           11 Jan 17 07:27 boot.b -> boot-menu.b
-rw-r--r--    1 root     root          728 Dec 17 07:53 chain.b
```

```
-rw-r--r--    1 root     root        12648 Nov 21 09:17 config-2.2.17
-rw-------    1 root     root        18432 Jan 17 07:27 map
-rw-r--r--    1 root     root          656 Dec 17 07:53 os2_d.b
-rwxr-xr-x    1 root     root      1042807 Nov 21 09:17 vmlinuz-2.2.17
```

En este caso se instalará la versión de lilo que ofrece un menú con las diferentes posibilidades en cuanto a SOs o configuraciones de sistema.

- Mediante "delay=20" indicamos cuánto tiempo esperará LILO hasta decantarse por la opción por defecto (especificada con "default=", más adelante). Este tiempo es en décimas de segundo, por lo que en nuestro caso esperaremos 2 segundos.
- Cada una de las opciones para sistemas Linux se configurará mediante "image=". Ahí concretaremos dónde reside el kernel para cada una de las imágenes. En este ejemplo existen dos imágenes Linux diferentes ("image=/vmlinuz" e "image=/vmlinuz.old") cada una con una versión diferente del kernel.
- Mediante "prompt" indicamos nuestro deseo a ser preguntados acerca de lo que hay que hacer, es decir, se nos deja elegir e incluso añadir opciones al arranque (ver BootPrompt-HOWTO para más detalles).
- Cuando queramos acceder a otro SO diferente a Linux desde LILO, deberemos utilizar "other=". En nuestro ejemplo indicamos que si queremos arrancar FreeBSD, LILO deberá acceder a /dev/hda1. Si quisiéramos que convivieran más SOs diferentes en nuestro sistema, bastaría con añadir más cláusulas "other=".
- Una nueva opción que incorporan las últimas versiones de LILO es "single-key". Si descomentamos esa línea y fijamos un alias para cada una de las opciones de arranque ("alias=1", por ejemplo), simplemente con pulsar una tecla elegiremos una opción de las posibles. Realmente esto sólo es cómodo si no utilizamos un menú, es decir, directamente desde el prompt de LILO.

```
Verifying DMI Pool Data ........

LILO boot:
FreeBSD        Linux
boot: Linux
Loading Linux.....................
Uncompressing Linux... Ok, booting the kernel.
```

Proceso de arranque del prompt en modo texto de LILO version 21.5-1 beta.

Un punto en el que me gustaría hacer hincapié es en el de la seguridad. Toda la seguridad del sistema puede verse afectada si LILO no está bien configurado. Imaginémonos que nos encontramos con un sistema que arranca con LILO y no está bien configurado. Solamente tecleando "linux single" en el prompt de LILO, podremos arrancar Linux como monousuario y de ahí acceder a ficheros importantes como los ficheros de claves (/etc/passwd, /etc/shadow) u otros confidenciales. Es más, una vez dentro del sistema, podríamos montar sistemas de ficheros de otros SOs y vulnerar también su seguridad. Todo esto se puede evitar si tomamos unas precauciones básicas.

En el anterior fichero de configuración hay una opción comentada ("#password=tatercounter2000") que nos permite proteger a LILO mediante una contraseña. Si descomentamos esa opción, fijamos una contraseña y reinstalamos LILO, habremos evitado una intrusión mediante "Linux single" de forma rápida y contundente. Sólo se nos olvida un detalle: por defecto /etc/lilo.conf es accesible para todo el mundo, por lo que sería conveniente fijar sus permisos a 400 (sólo lectura para el propietario, root), para evitar complicaciones.

```
argon:~# chmod 400 /etc/lilo.conf
argon:~# ls -al /etc/lilo.conf
-r--------    1 root     root         3806 Nov 30 12:12 /etc/lilo.conf
```

Para quienes no queráis pelearos con el fichero de configuración, existen varios asistentes y herramientas que nos harán el trabajo un poco menos pesado, aunque todo lo explicado con anterioridad sirve como guía para responder las preguntas que nos formulará el asistente. He aquí unos ejemplos:

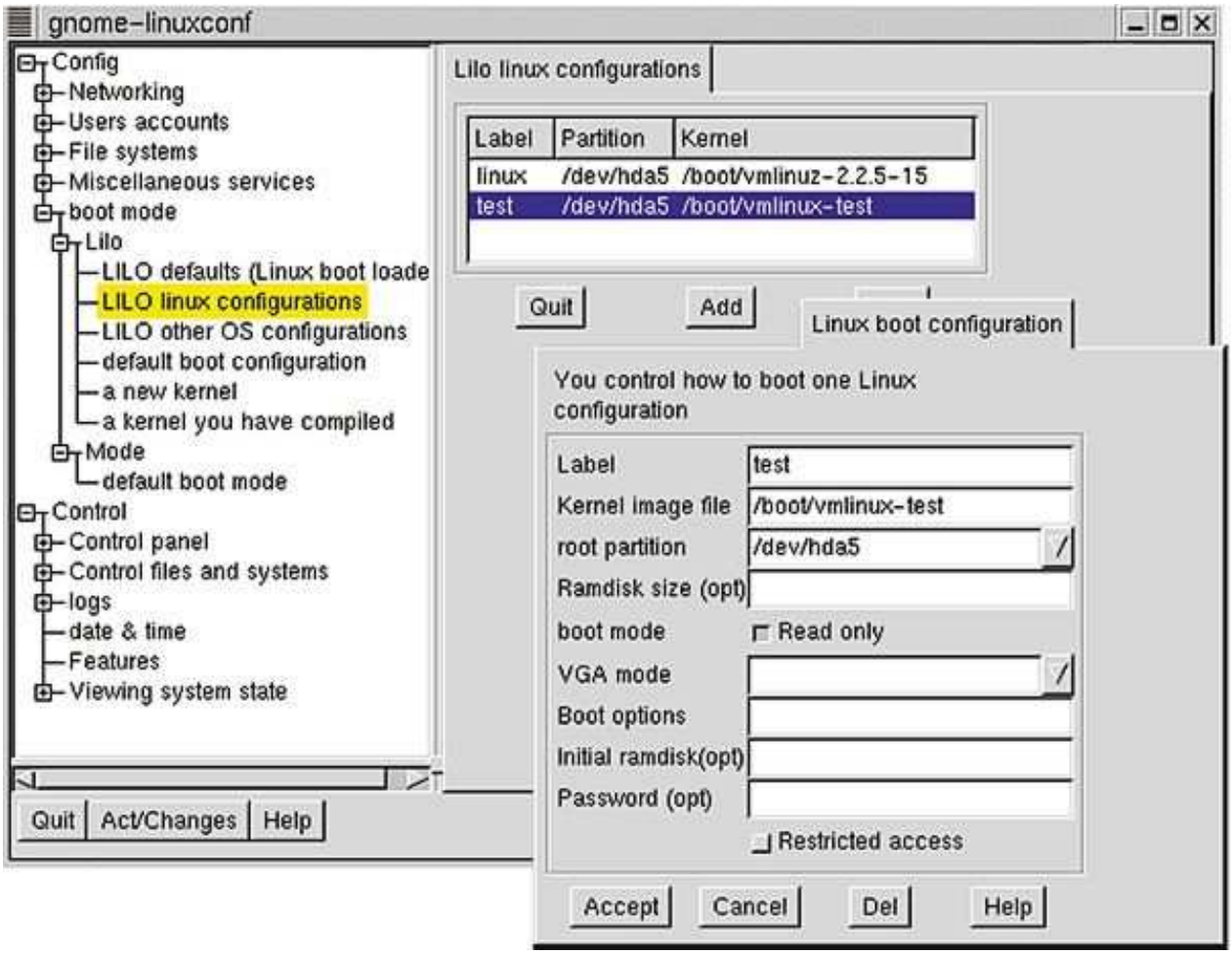

Figura 4. Configuración de LILO mediante asistentes de forma sencilla y eficaz.

En el aspecto meramente lúdico, LILO permite crear un menú gráfico más atractivo que un simple prompt en modo texto. Es posible hacer verdaderas virguerías si disponemos de tiempo y paciencia, como podéis observar en "Lilo Splash" (ver referencias al final del artículo).

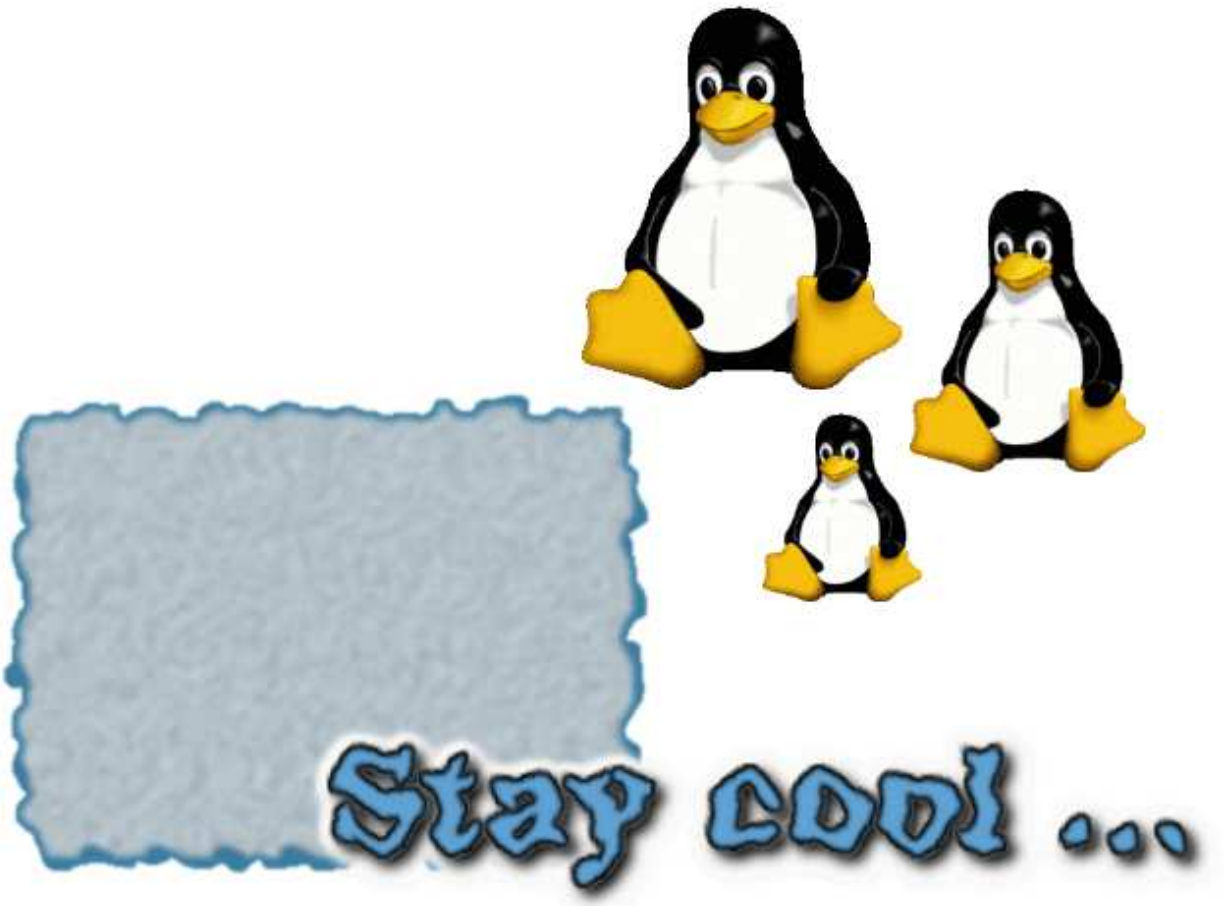

Figura 5. LILO puede configurarse para trabajar en modo gráfico y personalizar el menú mostrado de forma artística.

¿Qué hacer ante un desastre?

Al instalar LILO por primera vez, se guarda una copia de seguridad del sector de arranque en /boot/boot.xxyy (los números xxyy están en función del dispositivo físico del que se trate, para /dev/hda es típicamente 0300), por lo que siempre será posible volver al estado anterior a la instalación de LILO. Cuando reinstalemos lilo (ejecutando /sbin/lilo) se intentará guardar otra vez una copia del sector de arranque que va a ser sobreescrito, pero debemos ser precavidos porque si ya existía el fichero boot.xxyy, no se sobreescribe. Así, si nunca hemos modificado el nombre de nuestro primer fichero "boot.0300", por ejemplo, éste contendrá el sector de arranque antes de la primera instalación de LILO, no el inmediatamente anterior al que está ahora instalado.

La forma más común a la que muchos estamos acostumbrados de librarnos de LILO es utilizando el FDISK.EXE de DOS así:

```
A:\FDISK /MBR
```

Esto realmente es una chapuza que puede hacer inaccesibles las particiones no-DOS que tengamos en nuestro sistema. Una forma más elegante de desinstalar LILO es empleando el fichero que hemos comentado anteriormente, recuperando el sector de arranque original guardado en él. Si LILO se instaló en /dev/hda, podremos desinstalarlo de forma limpia así:

```
dd if=/boot/boot.0300 of=/dev/hda bs=446 count=1
```

Con ello copiamos los 446 primeros bytes de boot.0300 al sector de arranque del primer disco IDE (/dev/hda). El hecho de copiar únicamente los 446 primero bytes es para evitar sobreescribir la tabla de particiones actual y permitir acceder a las particiones que han sido creadas después de realizarse esa copia de seguridad del sector de arranque. Para otras configuraciones variarán el fichero de entrada (if=/boot/boot.0300) y el dispositivo destino o de salida (of=/dev/hda).

Conclusión

A lo largo del presente artículo hemos conocido cómo se produce el proceso de arranque en nuestros PC's y cómo configurar LILO para gestionarlo. Hemos demostrado parte de la gran potencia de LILO y sus opciones para aumentar la seguridad del sistema en sus dos aspectos: seguridad ante intrusos y seguridad ante problemas ("security and safety"). Una vez más os insto a seguir profundizando en el tema con la lectura de los HOWTO's relacionados con LILO y las páginas de manual (man lilo) ;-)

Para saber más...

- LILO mini HOWTO: "http://www.linuxdoc.org/HOWTO/mini/LILO.html
- Multiboot with LILO mini HOWTO: http://www.linuxdoc.org/HOWTO/mini/Multiboot-with-LILO.html
- LILO y NT Loader COMO: http://lucas.hispalinux.es/COMO-INSFLUG/COMOs/Linux-NT-Loader
- Bootdisk HOWTO: http://www.metalab.unc.edu/LDP/HOWTO/Bootdisk-HOWTO/index.html
- Bootdisks para Linux/NT/Win/DOS/Mac... : http://www.bootdisk.com
- LILO Splash, gráficamente artístico : http://www.13thfloor.at/Software/lilo-splash/Example/#liloconf
- Trinux : http://trinux.sourceforge.net

Este documento ha sido escrito por un miembro de e-GHOST, y su contenido es libre de ser reproducido en otros medios bajo las condiciones de la Licencia FDL (GNU Free Documentation License).