Arranque

Ok, ahora que tiene Slackware instalado en su sistema, debería aprender exactamente, qué controla la secuencia de arranque de su máquina y cómo arreglarla si se arruina de alguna manera. Si usa linux por harto tiempo, tarde o temprano cometerá un error que estropeará su cargador de arranque. Afortunadamente, esto no requiere una reinstalación para ser arreglado. Al contrario de muchos otros sistemas operativos que esconden los detalles subyacentes de su funcionamiento, Linux (y en particular Slackware) le da un control completo sobre el proceso de arranque. Simplemente, editando un par de archivos de configuración y re-ejecutando el instalador del cargador de arranque, puede, fácil y rápidamente cambiar(o romper) su sistema. Slackware incluso hace fácil el contar con un arranque dual con otros sistemas operativos tales como otras distribuciones Linux o Microsoft Windows.

mkinitrd

Antes de ir más lejos, se justifica una rápida discusión acerca del kernel Linux. Slackware Linux, incluye al menos dos (aunque a veces más) kernels diferentes. Aunque estos son compilados desde las mismas fuentes (y por lo tanto son los mismos), no son idénticos. Dependiendo de su arquitectura y la versión de Slackware, el instalador puede haber cargado varios kernels en su sistema. Los hay para sistemas mono-procesador, y para multi-procesador (en Slackware de 32-bits). Antiguamente, había muchos tipos de kernels distintos para ser instalados, con diversos tipos de controladores de disco duro. Lo que es más importante para nuestra discusión, es que hay kernels *“huge”* y *“generic”*.

Si observa dentro de su directorio `/boot` verá los distintos kernels instalados en su sistema.

```
darkstar:~# ls -1 /boot/vmlinuz*
/boot/vmlinuz-huge-2.6.29.4
/boot/vmlinuz-generic-2.6.29.4
```

Acá puede notar que hay dos kernels instalados: `vmlinuz-huge-2.6.29.4` y `vmlinuz-generic-2.6.29.4`. Cada versión de Slackware incluye a su vez, diferentes versiones de kernels y en ocasiones con nombres ligeramente distintos, así que no se alarme si lo que ve no corresponde exactamente con lo que se ha listado anteriormente.

Los kernels *huge* (enormes) son exactamente lo que podría pensar; son enormes, sin embargo, eso **NO** significa que tienen todos los controladores posibles compilados en si mismos. En vez de eso, estos kernels están hechos para arrancar (y ejecutarse) en cada computador concebible en que Slackware está soportado (puede haber unos pocos donde no arranque/funcione). Contienen soporte para hardware que su máquina no (ni nunca) tendrá, pero eso no debería importarle. Estos kernels son incluidos por varias razones, pero probablemente la más importante es su uso por parte del instalador de Slackware (estos son los kernels que se ejecutan en los discos de instalación).
Si elige dejar que el instalador configure el cargador de arranque por usted, éste usa estos kernels debido a la gran variedad de hardware que soportan.
En contraste, los kernels *generic*(genéricos) soportan muy poco hardware sin el uso de módulos externos. Si quiere usar alguno de esos kernels *generic*, deberá ocupar lo que se denomina *initrd* el cual es creado usando el programa ***mkinitrd***(8).

Entonces, ¿por qué debería usar un kernel *generic*? Actualmente el equipo de desarrollo de Slackware

recomienda el uso de un kernel *generic* por una variedad de razones. Tal vez la más obvia es el tamaño. Los kernels *huge* son aproximadamente el doble de grandes que un kernel *generic* antes de ser descomprimidos y cargados en memoria. Si está usando una máquina antigua o una con poca cantidad de RAM, podrá apreciar el ahorro que ofrece el *generic*. Otras razones son algo más difíciles de cuantificar. A veces pueden existir conflictos entre controladores incluidos en el kernel *huge* y, hablando en términos génerales, el rendimiento de éstos no es tan bueno como un *generic*. Además, al usar un kernel *generic* se pueden pasar argumentos especiales a los controladores de hardware de manera separada en vez de que estas opciones sean ingresadas a través de la línea de comandos. Algunas de las herramientas incluidas en Slackware funcionan mejor si el kernel usa algunos controladores como módulos en vez de estar estáticamente compilados. Si tiene problemas entendiendo esto, no se alarme, sólo piense: *"huge kernel = bueno, generic kernel = mejor"*.

Desafortunadamente, usar un kernel *generic* no es tan sencillo como usar un kernel *huge*. Para que arranque el sistema, debe incluir un par de módulos básicos en un initrd.
Los módulos son partes compiladas de código del kernel que pueden ser insertados o removidos en un kernel en ejecución (idealmente usando ***modprobe***(8)). Esto hace que el sistema sea un poco más flexible con el costo de una pequeña complejidad extra. Podría ser más fácil el considerar a los módulos como controladores de dispositivos, al menos para esta sección.
Típicamente, necesitará agregar el módulo para el sistema de archivos que eligió usar para su partición raíz durante el proceso de instalación, y, si ésta está localizada en un disco SCSI o un arreglo RAID, también se deberán agregar esos módulos. Finalmente si está usando RAID por software, encriptación de disco o LVM, se deberá crear un initrd independientemente de si usa un kernel *generic* o no.

Un initrd es un archivo comprimido ***cpio***(1), por lo que crear uno, no es una tarea simple. Afortunadamente, Slackware incluye una herramienta que facilita este proceso.
mkinitrd. Una discusión completa de ***mkinitrd*** está un poco fuera del alcance de este libro, pero de todas formas se mostrarán sus partes más importantes. Para una explicación más completa, revise el manual o ejecute ***mkinitrd*** con el argumento *–help*

```
darkstar:~# mkinitrd --help
mkinitrd crea un initial ramdisk (en realidad un archivo initramfs
cpio+gzip)
usado para cargar módulos del kernel que son necesarios para montar el
sistema de archivos de
la partición raíz, u otros módulos que pudiesen ser necesarios antes de que
ésta esté disponible.
Otros binarios pueden ser agregados al initrd y el script es fácil de
modificar.  Se creativo :-)
.... muchas más líneas eliminadas....
```

Cuando use ***mkinitrd***, necesitará contar con un poco de información previa: su partición raíz, el sistema de archivos que ésta usa, cualquier otro controlador de disco duro que se requiera, y si usará o no LVM, RAID por software y/o encriptación de disco. A menos que use algún tipo de controlador SCSI(y que la partición raíz esté localizada en este disco), sólo deberá conocer su partición raíz y el tipo de sistema de archivos de ésta.
Asumiendo que ha arrancado la instalación de Slackware usando un kernel *huge*, puede encontrar esta información con el comando ***mount*** o viendo el contenido de `/proc/mounts`.

```
darkstar:~# mount
/dev/sda1 on / type ext4 (rw,barrier=1,data=ordered)
```

```
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
usbfs on /proc/bus/usb type usbfs (rw)
/dev/sda2 on /home type jfs (rw)
tmpfs on /dev/shm type tmpfs (rw)
```

En el ejemplo provisto, puede ver que la partición raíz está localizada en `/dev/sda1` y es de tipo ext4. Si se quiere crear un initrd para este sistema, sólo se debe entregar esta información a ***mkinitrd***.

```
darkstar:~# mkinitrd -f ext4 -r /dev/sda1
```

Nótese que en la mayoría de los casos, ***mkinitrd*** es lo suficientemente listo para determinar esta información por su cuenta, pero nunca está demás especificarlo manualmente. \\Ahora que se ha creado el initrd, hace falta simplemente decirle a LILO dónde encontrarlo. La siguiente sección se encarga de eso.

Buscar todas las opciones de ***mkinitrd*** o peor aún, memorizarlas puede ser un verdadero dolor de cabeza, especialmente si prueba constantemente varios kernels. Esto se volvió tedioso para el equipo de desarrollo de Slackware, así que inventaron un archivo simple de configuración `mkinitrd.conf`(5). Puede encontrar un ejemplo localizado en el directorio `/etc/mkinitrd.conf.sample` y que puede ser fácilmente modificado para su sistema. El mio es:

```
darkstar:~# >/prompt>cat /etc/mkinitrd.conf.sample
# See "man mkinitrd.conf" for details on the syntax of this file
#
SOURCE_TREE="/boot/initrd-tree"
CLEAR_TREE="0"
OUTPUT_IMAGE="/boot/initrd.gz"
KERNEL_VERSION="$(uname -r)"
#KEYMAP="us"
MODULE_LIST="ext3:ext4:jfs"
#LUKSDEV="/dev/hda1"
ROOTDEV="/dev/sda1
ROOTFS="ext4"
#RESUMEDEV="/dev/hda2"
#RAID="0"
LVM="1"
#WAIT="1"
```

Para una descripción completa de cada una de estas líneas y qué es lo que hacen, deberá consultar el manual para `mkinitrd.conf`.
Copie el archivo de ejemplo a `/etc/mkinitrd.conf` y edítelo como más le acomode. Una vez que está configurado apropiadamente, sólo deberá ejecutar el comando ***mkinitrd*** con el argumento *-F*. Se construirá e instalará un initrd apropiado sin que tenga que recordar todos esos obscuros argumentos.

Si no está seguro de qué opciones especificar en el archivo de configuración o en la línea de comandos, hay una última alternativa. Slackware incluye una ingeniosa aplicación que indica que opciones son requeridas para el kernel que está ejecutando actualmente ***/usr/share/mkinitrd/mkinitrd_command_generator.sh***. Cuando ejecuta este comando, el

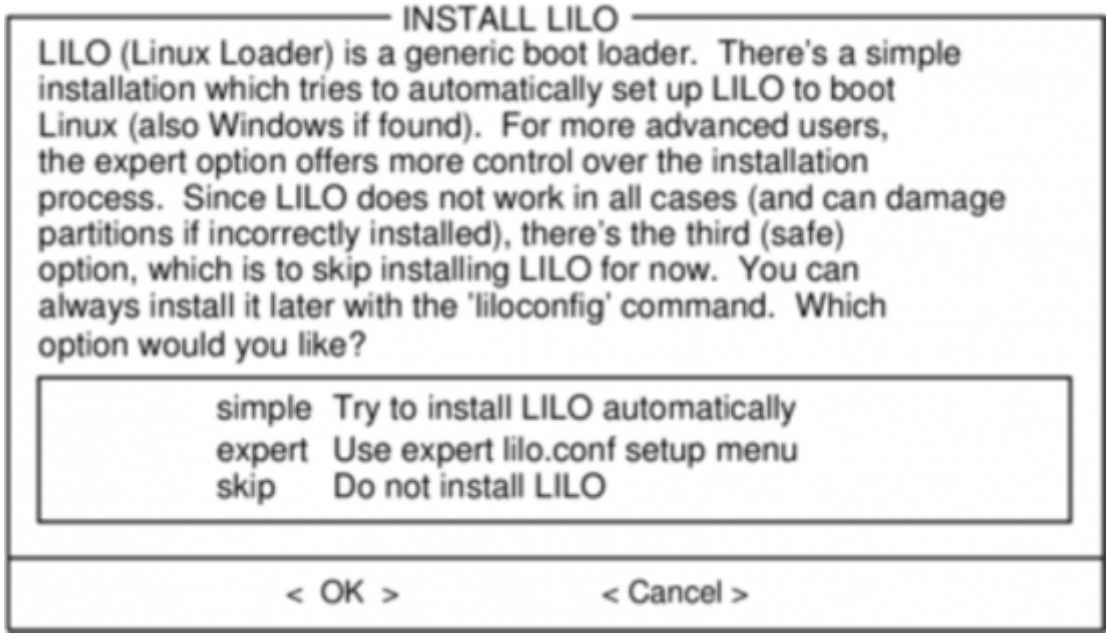
resultado es una línea de comando que puede ejecutar y que debería funcionar para su computador pero no está demás revisarla

```
darkstar:~# /usr/share/mkinitrd/mkinitrd_command_generator.sh
mkinitrd -c -k 2.6.33.4 -f ext3 -r /dev/sda3 -m \
  usbhid:ehci-hcd:uhci-hcd:ext3 -o /boot/initrd.gz
```

LILO

LILO es un acrónimo para *LInux LOader* y actualmente es el cargador de arranque por defecto instalado por Slackware Linux. Si ha usado antes otras distribuciones Linux, puede que esté más familiarizado con GRUB. Si prefiere usar GRUB, lo puede encontrar en el directorio `extra/` en uno de sus CDs (o DVD) de Slackware. Sin embargo como LILO es el cargador predeterminado que usa Slackware, nos enfocaremos exclusivamente en él.

Configurar LILO puede ser un poco desalentador para los nuevos usuarios, así que Slackware trae una herramienta especial de configuración llamada ***liloconfig***. Normalmente ***liloconfig*** es ejecutado por primera vez, por el instalador pero puede ejecutarlo en cualquier momento desde una terminal.



liloconfig tiene dos modos de operación: simple y experto.
El modo *"simple"* intenta configurar LILO automáticamente por Usted. Si Slackware es el único sistema operativo instalado en su computador, el modo *"simple"* casi siempre hará lo correcto fácil y rápidamente. También es muy bueno en detectar instalaciones de Windows y agregarlas a `/etc/lilo.conf`, así puede elegir que sistema operativo arrancar al prender su computador.

Para usar el modo experto, necesita saber la partición raíz de Slackware. También puede configurar otros sistemas operativos Linux si conoce sus respectivas particiones raíz, pero esto puede no funcionar tan bien como espera. ***liloconfig*** intentará arrancar cada sistema operativo Linux con el kernel de Slackware y esto probablemente no es lo que quiere. Afortunadamente, el configurar particiones de Windows en modo experto es trivial.
Un consejo cuando use el modo experto: casi siempre deberá instalar LILO en el *Master Boot Record* (MBR). Hace mucho tiempo, era recomendado instalar el cargador de arranque en la partición raíz y

establecer ésta como arrancable (bootable). Hoy en día, LILO ha madurado mucho y es seguro el instalarlo en el MBR, de hecho, encontrará menos problemas si lo hace.

liloconfig es un buen método para configurar rápidamente su cargador de arranque, pero si quiere saber realmente qué es lo que hace, deberá ver el archivo de configuración: `lilo.conf`(5) en el directorio `/etc`. `/etc/lilo.conf` está dividido en varias secciones. En la parte superior, encontrará una sección *"global"* donde puede especificar cosas como dónde quiere que se instale LILO (generalmente en el MBR), cualquier imagen especial para mostrar al arrancar, y el tiempo de espera después de el cual LILO iniciará el sistema operativo predeterminado. Aquí está en parte, cómo luce la sección global de mi lilo.conf

```
# LILO configuration file

boot = /dev/sda
  bitmap = /boot/slack.bmp
  bmp-colors = 255,0,255,0,255,0
  bmp-table = 60,6,1,16
  bmp-timer = 65,27,0,255

append=" vt.default_utf8=0"
prompt
timeout = 50

# VESA framebuffer console @ 1024x768x256
vga = 773
.... many more lines ommitted ....
```

Para una lista completa de todas las opciones posibles de LILO, deberá consultar el manual de `lilo.conf`. En este documento, discutiremos brevemente las opciones más comunes.

La primera cosa que debería llamar su atención es la línea *"boot"*. Ésta determina dónde se instala el cargador de arranque. Para instalarlo en el MBR de su disco duro, simplemente liste en esta línea la entrada del dispositivo de disco duro que desee. En mi caso estoy usando un disco duro SATA que se muestra como un dispositivo SCSI `/dev/sda`. Para instarlo en el bloque de arranque de una partición, tendrá que colocar la entrada de dispositivo de la **partición**; por ejemplo si lo instala en la primera partición del único disco duro SATA en su computador, entonces probablemente usará `/dev/sda1`.

La opción *"prompt"* simplemente le dice a LILO que pregunte (prompt) qué sistema operativo arrancar. Éstos son listados más adelante en el archivo, en su propia sección, ya se llegará a eso.
La opción *"timeout"* le indica a LILO cuánto debe esperar (en décimas de segundo) antes de arrancar el sistema operativo predeterminado. En mi caso es de cinco segundos. Parece ser que algunos sistemas demoran bastante en mostrar la pantalla de pantalla de arranque por lo que pueden necesitar un tiempo mayor al que mostré. Esto es en parte, por qué LILO utiliza un tiempo bastante largo para *"timeout"* (alrededor de dos minutos).
La línea *"append"*, en mi caso fue configurada por ***liloconfig***. Puede (y probablemente debería) ver algo similar cuando revise su propio `/etc/lilo.conf`. No entraré en detalles de por qué es necesaria, así que sólo deberá confiar en que algunas cosas funcionan mejor si está presente. :^)

Ahora que ya se ha visto la sección global, es hora de revisar la parte de los sistemas operativos. Cada entrada Linux, comienza con una línea *"image"*. Los sistemas operativos Windows son especificados mediante una línea *"other"*. Un ejemplo de `/etc/lilo.conf` que puede arrancar

Linux o windows es el siguiente:

```
# LILO configuration file
... global section ommitted ....
# Linux bootable partition config begins
image = /boot/vmlinuz-generic-2.6.29.4
  root = /dev/sda1
  initrd = /boot/initrd.gz
  label = Slackware64
  read-only
# Linux bootable partition config ends
# Windows bootable partition config begins
other = /dev/sda3
  label = Windows
  table = /dev/sda
# Windows bootable partition config ends
```

Para sistemas operativos Linux como Slackware, la línea *"image"* especifica qué kernel cargar. En este caso, se usará un kernel 2.6.29.4 localizado en `/boot/vmlinuz-generic-2.6.29.4`.
Las secciones restantes son bastante claras, le dicen a LILO dónde encontrar el sistema de archivos raíz, qué initrd (si hay) usar, y si el sistema de archivos se monta inicialmente como sólo lectura. La línea del initrd es muy importante si usa un kernel *generic*, LVM o un RAID por software ya que indica a LILO dónde encontrar el initrd creado usando ***mkinitrd***.

Una vez que ha configurado `/etc/lilo.conf` acorde a su máquina, simplemente ejecute ***lilo***(8) para instalarlo. Al contrario de GRUB y otros cargadores de arranque, LILO requiere que se ejecute ***lilo*** cada vez que se haga un cambio en su archivo de configuración o si no la nueva (modificada) imagen del cargador de arranque no será instalada y las modificaciones no tendrán efecto.

```
darkstar:~# lilo
Warning: LBA32 addressing assumed
Added Slackware *
Added Backup
6 warnings were issued.
```

No se asuste por las advertencias que pueda ver cuando ejecute ***lilo***. A menos que vea un error fatal, las cosas deberían funcionar bien. En particular, el problema que aparece con *LBA32 addressing* es común.

Arranque dual

Un cargador de arranque (como LILO) es una cosa muy flexible, ya que sólo existe para determinar qué disco duro, partición, e incluso versión de kernel arrancar. Esto sugiere inherentemente una elección previa al arranque, así que la idea de tener más de un sistema operativo aparece de manera natural para un usuario de LILO (o GRUB).

La gente cuenta con un *"arranque dual"* por varias razones: algunos quieren tener una instalación estable de Slackware en una partición o unidad y un *"sandbox"* de desarrollo en otra; también se podría querer tener Slackware en una partición y en otra, una distribución de Linux o BSD; incluso

otras personas pueden tener Slackware en una partición y en una distinta, algún sistema operativo propietario (por trabajo o por alguna aplicación que no esté disponible en Linux).

El arranque dual no debe ser tomado a la ligera ya que usualmente implica que habrá dos (o más) sistemas operativos intentando administrar el cargador de arranque. Si desea contar con ésta característica, hay una alta probabilidad de que uno de los sistemas operativos modifique o sobrescriba las entradas del cargador de arranque sin su intervención directa. Si esto ocurre, deberá modificar LILO or GRUB manualmente para poder iniciar cada sistema operativo.

Hay dos maneras para realizar un arranque dual (o múltiple): puede colocar cada sistema operativo en su propio disco duro (común en un computador de escritorio dónde se tiene el lujo de tener más de una bahía de unidad); o bien cada sistema operativo tiene su propia partición (común en un computador portátil dónde sólo hay presente una unidad física de disco duro).

Arranque dual con particiones

Para configurar un sistema con arranque dual, con cada sistema operativo en su propia partición, primero debe crear éstas. Esto es más fácil si se realiza previo a instalar el primer sistema en cuyo caso es sólo un simple caso de planear y dividir su disco duro como le sea conveniente. Vea la sección llamada “Partitioning” para información en el uso de las aplicaciones de particionamiento ***fdisk*** o ***cfdisk***

Si va a hacer un arranque dual con dos distribuciones Linux, no es aconsejable el intentar compartir un directorio /home entre éstas. Aunque es técnicamente posible, el hacerlo incrementa las posibilidades de que sus configuraciones personales sean dañadas por entornos de escritorio competidores o distintas versiones de éstos.

Sin embargo es seguro el compartir una partición swap.

Deberá particionar su disco en al menos tres partes:

- Una partición para Slackware
- Una partición para el segundo sistema operativo
- Una partición swap

Primero, instale Slackware Linux en la primera partición del disco duro como describe en Capítulo 2, Instalación.

Después de que Slackware ha sido instalado, iniciado y que ha confirmado que todo funciona como debería, entonces reinicie e instale el segundo sistema. Éste siempre ofrecerá usar todo el disco; obviamente esto **no** es lo que se quiere, así que restrínjalo sólo a la segunda partición. Más aun, intentará instalar un cargador de arranque en el comienzo del disco duro sobrescribiendo LILO.

Con respecto al cargador de arranque, tiene unos cuantos cursos de acción posibles:

Si el segundo sistema es Linux, no le permita instalar un cargador de arranque.

Si está haciendo un arranque dual con otra distribución Linux, el instalador de élla a menudo le pide instalar un cargador. Ciertamente es libre de no hacerlo y manejar la distribución y Slackware con

LILO.

Dependiendo de la distribución, podría tener que editar LILO más frecuentemente que si sólo ejecutase Slackware. Algunas distribuciones son notorias por sus actualizaciones frecuentes del kernel lo que significa que tendrá que editar LILO después de cada una de ellas para reflejar los nuevos cambios. Pero si no quisiese editar archivos de configuración de vez en cuando, probablemente no hubiese elegido Slackware.

Si el segundo sistema operativo es una distribución Linux, deje que reemplace LILO por GRUB

Si está usando un arranque dual con otra distribución Linux, entonces todo es capaz de funcionar perfectamente usando GRUB en vez de LILO o bien puede instalar Slackware al final y usar LILO para manejar los sistemas. Ambos, LILO y GRUB, tienen buenas capacidades de detección así que cualquiera que sea instalado al final, debería detectar la presencia de la otra distribución y crear una entrada para ésta.

Ya que otras distribuciones a menudo intentan actualizar sus menús de GRUB, siempre existe la posibilidad de que algo pueda corromperse y que de pronto no pueda iniciar Slackware. Si esto ocurre, no entre en pánico; sólo arranque la otra distribución Linux y edite GRUB manualmente de manera tal que apunte a la partición, kernel e initrd (si usa) correctos para Slackware en la entrada del menú.

Permita que el segundo sistema operativo sobrescriba LILO, y luego vuelva a reinstalar y reconfigurar LILO manualmente

Ésta no es una mala elección, sobre todo cuando el sistema operativo secundario es Windows. Pero los riesgos potenciales son que, cuando Windows se actualice podría intentar sobrescribir nuevamente el MBR (master boot record) y así tendría que reinstalar LILO de nuevo manualmente.

Para restablecer LILO luego que otro sistema operativo lo haya borrado puede arrancar desde su medio de instalación de Slackware y entrar en la etapa de configuración. <emphasis>No</emphasis> reparticione su disco ni reinstale Slackware, salte inmediatamente a la sección Configurar,

Incluso si está usando la opción *“simple”* para la instalación, LILO debería detectar ambos sistemas operativos y configurar un menú razonable por Usted. Si falla entonces deberá agregar las entradas por su cuenta.

Arranque dual desde discos duros

El arranque dual entre diferentes discos duros físicos, es a menudo más sencillo que cuando se trata de particiones ya que la BIOS o EFI del computador casi siempre cuenta con un selector de dispositivo de arranque que permite interrumpir este proceso inmediatamente después de *POST* y elegir que unidades deben tener más prioridad

La tecla para entrar en el selector de arranque es diferente para cada marca de placa madre; consulte su manual o ponga atención a la pantalla de bienvenida al iniciar su computador para saber cuál es la correcta. Típicamente son: F1, F12, DEL. Para computadores Apple, siempre es la tecla

Option (Alt)

Si administra la prioridad de arranque vía la BIOS o EFI, entonces cada cargador de arranque en cada disco duro, sólo es consciente de su propia unidad y nunca interferirá con otra. Esto es algo contrario a lo que el cargador de arranque está diseñado para hacer, pero puede ser una buena solución alternativa, cuando se trata con sistemas operativos propietarios que insisten en ser los únicos en el sistema en detrimento de la elección del usuario.

Si no tiene múltiples discos duros internos y no se siente cómodo haciendo malabares con otras particiones y sistemas operativos en su computador, también podría considerar una unidad removible (como un pendrive) autoarrancable o incluso una máquina virtual que le dé acceso a otros sistemas operativos. Ambas opciones quedan fuera del alcance de esta guía pero son algo común y pueden ser la elección adecuada dependiendo de sus necesidades.

Navegación de capítulos

Capítulo previo: Instalación

Capítulo siguiente Comandos Básicos del Shell

Fuentes

- Fuente original: http://www.slackbook.org/beta

- Escrito originalmente por: Alan Hicks, Chris Lumens, David Cantrell, Logan Johnson
- Traducción original por: Mandrake

slackware, booting, mkinitrd, lilo, dual-boot, author jcourbis