

Instalación de sistemas Linux, DOS y Windows XP en una memoria USB

Por Gabriel.Verdejo.Alvarez@gmail.com revisado Junio'2009

La primera versión de este documento se publicó en Diciembre de 2007 bajo el título “**Instalación de Debian Linux y MS-DOS en una memoria USB**”. Este documento recoge el resultado de la corrección de algunos errores, la ampliación de funcionalidades y las sugerencias de varios lectores. Mi más sincero agradecimiento a todos vosotros.

Este escrito ha sido reestructurado en dos partes claramente diferenciadas que hacen referencia al procedimiento de instalación de los sistemas en la memoria Flash. El primero permite realizarlo mediante un procedimiento más sencillo y flexible, ideal para usuarios con prisa. La segunda parte recoge el procedimiento manual, y por tanto más laborioso, que permite el control total de las instalaciones realizadas.

Método 1: Instalación a través de una máquina virtual. La instalación de los distintos sistemas se realiza dentro de una máquina virtual. Esta opción permite el uso de las instalaciones semi-automáticas que proporcionan muchos sistemas operativos. Una vez concluida la personalización del sistema procedemos a volcar la imagen en la memoria Flash.

Método 2: Instalación tradicional directamente en la memoria USB. Cada uno de los sistemas elegidos se instala directamente sobre la memoria Flash. El usuario tiene un control directo del proceso y permite el uso de instalaciones personalizadas.

Las habilidades técnicas necesarias para seguir este tutorial son de nivel medio, ya que requieren unas ciertas nociones de Linux. De todas formas he intentado que cualquier usuario independientemente de su nivel pueda conseguir la instalación de los sistemas siguiendo las indicaciones y poniendo algo de su parte.

Y como ya sabemos, “*hace más quien quiere que quien puede*”.

1. Introducción

En este documento explicaremos cómo instalar diferentes sistemas operativos en una memoria Flash USB. Instalar un sistema operativo tipo Linux conjuntamente con uno basado en MS-DOS o Windows XP es finalmente posible de una forma razonablemente sencilla gracias a programas de virtualización como *VirtualBox* [www18].

La utilidad de este procedimiento se basa en aprovechar la capacidad que ya incorporan todos los ordenadores de sobremesa y portátiles de iniciar el sistema desde un dispositivo USB. Hoy en día las memorias Flash son muy baratas, de mucha capacidad y de tamaños muy reducidos, lo que nos permite disponer de un sistema portátil totalmente funcional.

En el caso de sistemas operativos basados en MS-DOS [www8] o PC-DOS [www9] la idea es poder iniciar el sistema de forma que podamos realizar operaciones que generalmente bajo Windows u otros sistemas operativos no podemos. Actualizar la BIOS de la placa base, pasar programas de test y recuperación de sistema como el genial Hiren's BootCD [www10] o realizar copias del disco duro son ahora tareas sencillas y eficientes.

El tamaño de las particiones DOS, en principio, puede ser el que nos permita el sistema de ficheros FAT¹ [www23] que utilicemos. Sin embargo con algunos modelos de portátiles he comprobado que con particiones inferiores a unos 40Mbytes es imposible iniciar el sistema MS-DOS. De esta manera recomiendo crear particiones de al menos 60Mbytes.

El espacio mínimo que se requiere para una instalación mínima funcional de Debian es de unos 400Mbytes. Todo y que es factible tenerla en una memoria de 256Mbytes, al instalarse el sistema requiere de más espacio de almacenamiento ya que necesitamos bajar los paquetes (.deb) y descomprimirlos antes de instalarlos. Una vez actualizado el sistema y eliminando los paquetes y programas extras no necesarios, podemos ir adaptando el sistema al espacio disponible.

2. Método 1: Instalación mediante una máquina virtual

En este primer apartado realizaremos la instalación de los sistemas operativos mediante el uso de una máquina virtual. Actualmente hay varias opciones tanto libres como de pago que nos permiten realizar este proceso de una forma sencilla. Nuestro único requerimiento es que la máquina virtual cuente con soporte USB para poder acceder a nuestra memoria Flash.

Para este documento he utilizado *VirtualBox* [www18] que dispone de versiones gratuitas tanto para Windows como para Linux o MacOS X. La idea fundamental es la de simular que la Flash es el disco duro de una máquina y por tanto realizar las instalaciones como en nuestro propio PC.

Una vez personalizado el sistema y realizadas las comprobaciones pertinentes, volcaremos la información a la memoria USB. Este sistema tiene dos grandes ventajas que lo hacen ideal para usuarios noveles o que no desean complicarse en exceso:

- *Es fácil de realizar*: El proceso es análogo a la instalación en cualquier PC, con lo que podemos utilizar los CDs y programas de instalación que proporcionan los sistemas operativos.

Como trabajamos en una máquina virtual en caso de equivocarnos o realizar cualquier catástrofe no afecta a nuestro sistema principal, de forma que podemos repetir el proceso tantas veces como sea necesario.

- *No necesitamos guardar copias*: La propia máquina virtual contiene el sistema original, con lo que podemos modificarlo o actualizarlo tanto como deseemos. Simplemente volvemos a volcarlo en la memoria Flash y listo para utilizar.

¹ El sistema usual FAT16 permite hasta 4Gbytes.

2.1. Creación de una máquina virtual con el espacio de disco adecuado

Una vez instalado el software de máquina virtual en nuestro sistema anfitrión, debemos disponernos a configurar los parámetros básicos del sistema. En este ejemplo se repasa la configuración necesaria para un sistema VirtualBox. En caso de usar otro programa de virtualización deberíamos ajustar los valores de configuración los más parecido posible.

Es **importante** destacar que asumimos que ya tenemos disponibles los CDs o DVDs de instalación de los distintos sistemas que vamos a utilizar. En el caso de contar únicamente con las imágenes *.iso* podemos montarlas perfectamente en la máquina virtual ahorrándonos el soporte físico.

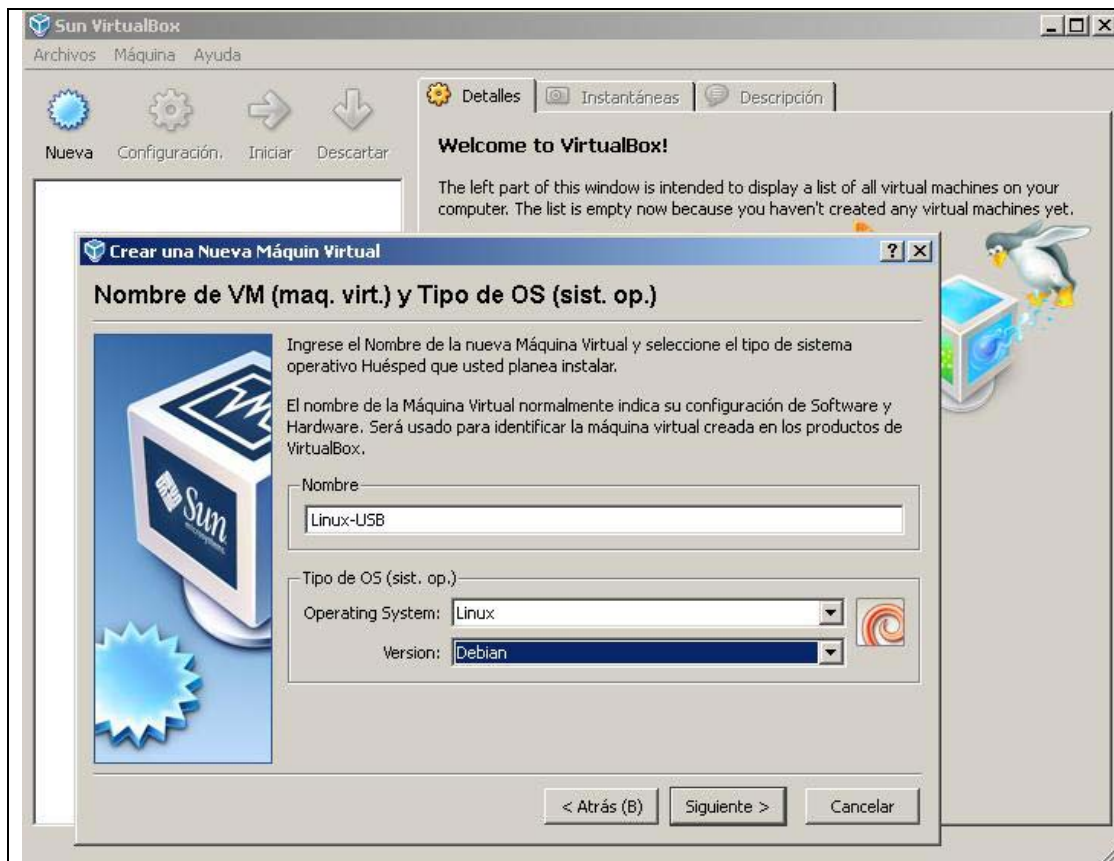


Figura 1: Creación de una nueva máquina virtual con VirtualBox.

Primero realizaremos la creación de una instancia nueva. Los valores que lleve por defecto suelen ser más que suficientes. Sin embargo sí hay tres aspectos a los que debemos prestar atención:

- *Acceso a los dispositivos USB:* Debemos asegurarnos que los dispositivos USB, y por tanto nuestra memoria Flash, son totalmente visibles desde la máquina virtual.
- *Espacio en disco:* Se nos obliga a definir un máximo de tamaño de disco, este **siempre ha de ser al menos** tan grande como nuestra memoria USB. Para evitar problemas siempre es mejor indicar un tamaño mayor al real, por ejemplo en una Flash de 8Gbytes podemos crear un disco de 8,1Gbytes.

- *Conectividad de red:* Todo y que no es estrictamente necesario, excepto para instalaciones que requieran acceso a Internet, es deseable que la máquina tenga acceso a nuestra red. Esto nos permite por ejemplo realizar instalaciones mínimas y bajar gradualmente lo que necesitemos.

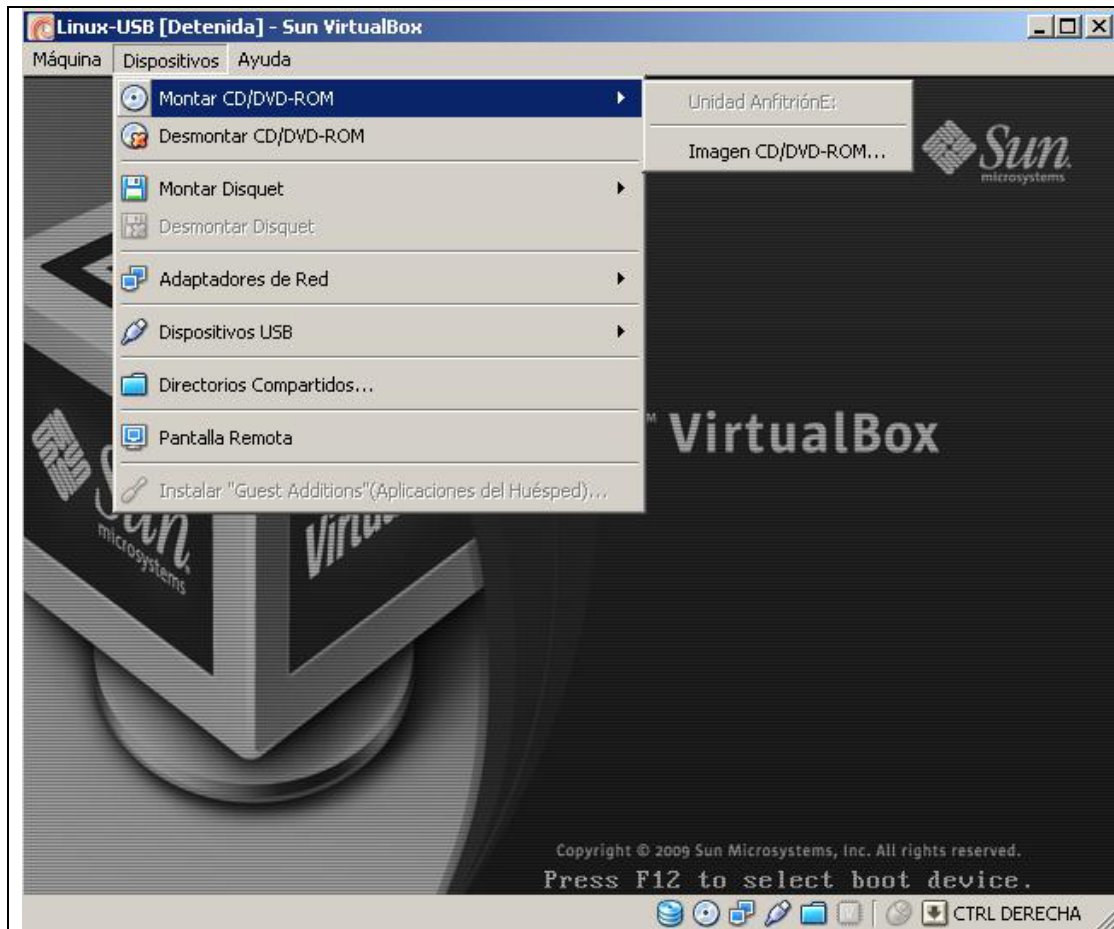


Figura 2: Selector de opciones para vincular dispositivo USB y CD-ROM.

Una vez definidas las características básicas de la máquina debemos configurar el disco virtual asociado. En el caso de VirtualBox nos deja elegir entre crear un fichero de tamaño fijo o variable. En nuestro caso probablemente sea mejor el de tamaño fijo ya que nosotros copiaremos toda la memoria USB y por lo tanto usaremos “todo” el espacio. Al estar ya creado nos ahorraremos el tiempo de que vaya aumentando el espacio del fichero conforme usemos más espacio durante las instalaciones. Por otro lado, es más probable que al crearlo de una vez esté más compactado, lo que mejora ligeramente el tiempo de acceso al disco.

A continuación necesitamos “copiar” la memoria USB al disco duro virtual. Este paso es **necesario** ya que para poder luego aprovechar el disco de la máquina virtual y que funcionen sin problemas, estos han de tener la misma geometría de disco. Los usuarios avanzados podrían optar por realizar un particionamiento de disco manual indicando los sectores de inicio y final, pero algunas veces puede que no funcione cuando lo pasen de nuevo a la memoria Flash.

Para realizar este paso podemos utilizar cualquier *live-cd* de Linux. En este ejemplo se utilizará el sistema de instalación por red (*netinst*) de Debian Lenny [www21] que ocupa “únicamente” unos 160Mbytes. En [www22] podemos encontrar una abundante recopilación de otras alternativas posibles.

Primero procederemos a montar la imagen ISO del sistema así como nuestra memoria USB en la máquina virtual. A continuación iniciaremos el nuevo sistema desde la imagen del CD de instalación. Nuestro objetivo es obtener un intérprete de comandos (Shell) que nos permita efectuar la copia de los datos. En el caso de Debian Lenny seleccionaríamos *Advanced Options* y a continuación *Graphical rescue mode*.

Una vez seleccionadas las opciones de idioma y configurada la red, seleccionamos la opción de “intérprete de órdenes (Shell) en el entorno del instalador”. Posteriormente verificamos los distintos dispositivos que ve nuestra máquina virtual mediante el comando *fdisk*.

```
# fdisk -l

Disk /dev/hda: 120.0 GB, 120034123776 bytes
255 heads, 63 sectors/track, 14593 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

Disk /dev/hda doesn't contain a valid partition table

Disk /dev/sda: 120.0 GB, 120034123776 bytes
255 heads, 63 sectors/track, 14593 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks    Id System
/dev/sda1 *          1           15    1199904+    c  W95 FAT32 (LBA)
```

Debemos observar dos discos distintos que corresponden al disco virtual (sin inicializar) y a la memoria USB que tenemos conectada. En nuestro ejemplo tenemos */dev/hda* como disco virtual y */dev/sda* como Flash. Una vez identificados los dispositivos procedemos a la copia de la Flash al disco virtual mediante el comando *cat* (es **muy importante** realizar este proceso con cuidado ya que de lo contrario la Flash quedará inutilizada y deberá ser inicializada ¡de nuevo!).

```
# cat /dev/sda > /dev/hda
```

Ahora disponemos de una copia (con la misma geometría) en el disco duro de nuestra máquina virtual. Ya podemos reiniciar el sistema y proceder a instalar los sistemas operativos. Si se desea también podemos particionar de nuevo o modificar las particiones existentes con programas tipo *fdisk*/Partition Magic² o similares.

2.2. Instalación de sistemas DOS (Hiren’s BootCD)

Para realizar la instalación de un sistema basado en DOS necesitaremos una imagen de arranque que contenga las utilidades *fdisk*, *format* y *sys*. Ejemplos de sistemas que podemos utilizar podrían ser los discos o CDs de las instalaciones de sistemas Windows 95/98 o el Hiren’s BootCD [www10] que es un recopilatorio de grandes utilidades de sistema.

¡Es **necesario** tener en cuenta que la partición DOS ha de ser la primera del disco ya que los sistemas Windows acceden únicamente a esta!

² Cabe destacar que podemos crear/borrar/modificar/formatear las particiones a nuestro gusto pero siempre sin modificar la geometría del disco.

En el caso de tener varios sistemas operativos la mejor opción es crear la primera partición de mayor tamaño y realizar en ella la instalación del sistema DOS. De esta forma cuando conectemos la Flash a otro ordenador siempre nos permitirá ver la partición con más espacio libre.

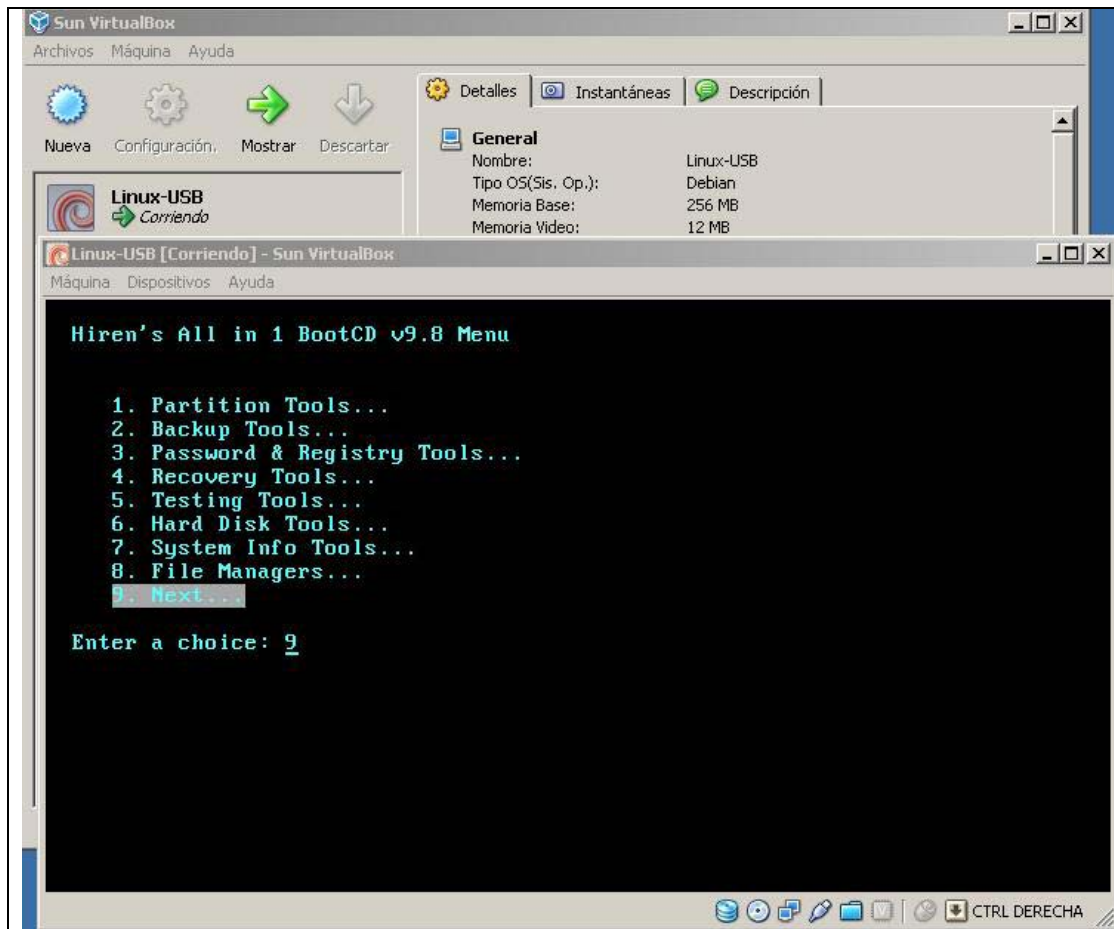


Figura 3: Hiren's BootCD instalado en el disco duro virtual.

Una vez iniciada la máquina virtual con la imagen o el CD de sistema debemos proceder a formatearla y añadirle el sistema de arranque. Podemos utilizar directamente el comando *format /s* para añadir el sistema de inicio a nuestra partición o formatearla y posteriormente instalarlo mediante el comando *sys*.

En el caso de desear instalar el Hiren's BootCD en la Flash, además, deberemos proceder a la copia de los archivos del menú de inicio así como los programas existentes.

```
xcopy /s a:\*.* c:  
xcopy /s r:\*.* c:
```

2.3. Instalación de sistemas Windows XP

La instalación de un sistema Windows XP en una memoria Flash plantea cuestiones importantes que nos obligarán a decidir aspectos críticos. Para empezar cabe señalar que **NO** es factible tener una instalación universal igual que el sistema instalado en el disco duro de un PC. El tamaño que ocuparía en una memoria USB y la imposibilidad de disponer de todos los drivers existentes para tarjetas de video, red, chipsets... hacen de esta tarea una misión imposible.

Sin embargo sí es posible realizar otras aproximaciones cercanas que nos permitan disfrutar de un sistema funcional sin renunciar excesivamente a nuestras pretensiones. Las dos opciones que analizaremos en este apartado son la instalación de un sistema mínimo de emergencia de Windows XP y la instalación clásica bajo una máquina virtual ejecutándose en Linux.

También debemos tener en cuenta que en el caso de instalar un sistema Windows XP en la memoria USB este debería encontrarse en la primera partición para ser visible desde otro sistema Windows. Es importante formatear la partición como FAT32 para poder disponer de nombres largos y particiones de más de 4 Gbytes.

Debido a que los procesos que comentaremos a continuación sobrescriben la memoria Flash directamente y crean una única partición, una vez comprobado que funcionan adecuadamente deberemos reparticionar la memoria USB encogiando la partición si deseamos compartir el espacio con Linux.

2.3.1 Instalación de un sistema Mini-Windows XP

El objetivo de este apartado es el de disponer de un sistema mínimo que ronde los 400Mbytes y nos permita ejecutar diferentes acciones de emergencia sobre un disco. Para realizar esta tarea podemos crear una imagen nueva a partir del CD de instalación de Windows XP mediante el entorno *Bart's Preinstalled Environment* también conocido como BartPE [www24] o utilizar una imagen ya creada y copiarla en la unidad Flash mediante el programa PeToUSB [www26].

Para crear un sistema personalizado necesitaremos un CD de instalación “original” de Windows XP, ya que a la hora de confeccionar la imagen BartPE necesita algunos archivos de instalación. Podemos utilizar cualquier versión de CD de Windows XP, lo que nos permite crear un sistema con los ServicePack aplicados evitando tener que retocar la imagen una vez creada con las actualizaciones críticas.

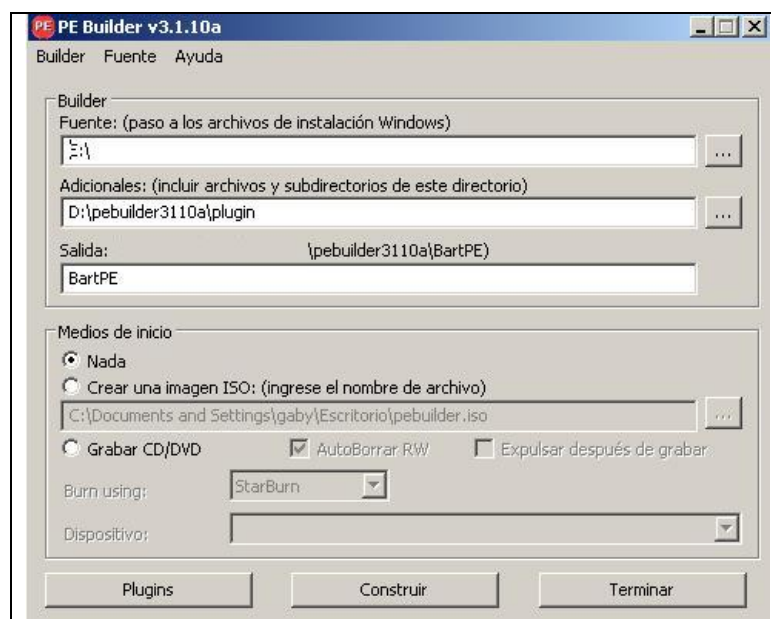


Figura 4: Creación de una imagen de Windows XP con BartPe.

Si deseamos añadir algunos programas extra o utilidades concretas podemos incluirlos en el directorio de *plugins* que nos crea el BartPE. En la bibliografía [www27] podemos encontrar una gran selección de programas (compresores, utilidades de gestión de disco...) que podemos añadir a nuestra imagen. Obviamente cuantos más plugins añadamos mayor tamaño tendrá la imagen resultante y por tanto más espacio de RAM consumirá al ejecutarse.

La segunda opción consiste en conseguir una imagen .iso ya creada (al igual que en el caso anterior deberemos disponer de la licencia correspondiente) y únicamente escribirla en la memoria USB. En este caso únicamente necesitamos utilizar la utilidad PeToUSB que nos permite grabar la imagen directamente en la memoria Flash.

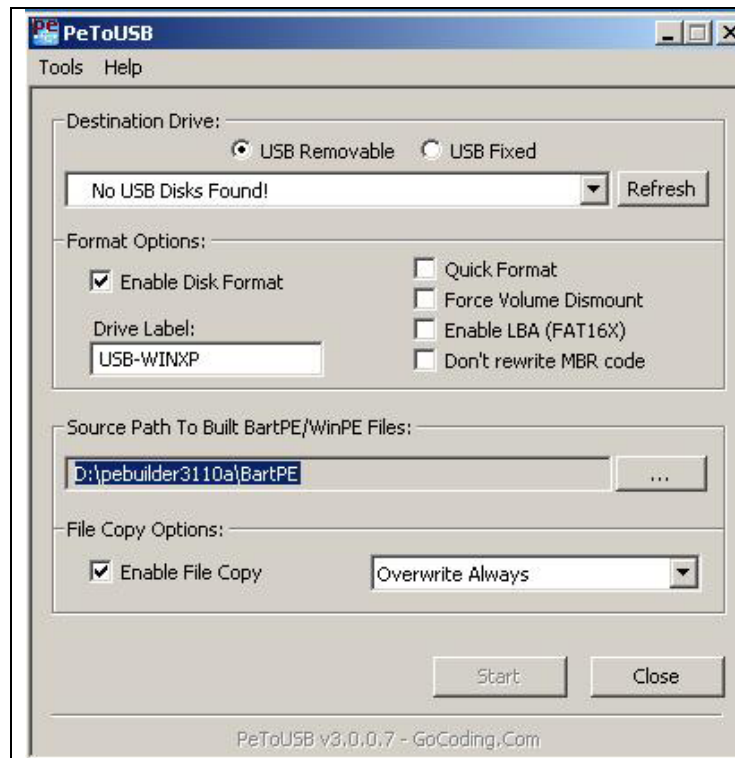


Figura 5: Instalación de imagen .iso en memoria Flash.

En caso de que el sistema nos formatee la unidad con FAT16 es muy recomendable pasarlo a FAT32, que entre otras ventajas tiene la de soportar nombres de archivos “largos”. Para ello podemos utilizar cualquiera de las utilidades que por ejemplo nos proporciona el Hiren’sBoot CD.

2.3.2 Windows XP Embedded

La propia Microsoft distribuye una versión de Windows XP que permite ejecutarse totalmente desde memoria RAM. El *Windows XP Embedded* [www28][www30] es una solución que puede contemplarse para casos concretos como el de portátiles de bajo coste tipo OLPC. En [www25] se puede encontrar un ejemplo práctico.

2.4. Instalación de sistemas Linux

La instalación de un sistema Linux es relativamente sencilla ya que podemos utilizar cualquier imagen o CD de instalación existente. A diferencia de los sistemas basados en DOS, la partición que contiene Linux puede ser la que deseemos. De esta forma es conveniente primero particionar la Flash, instalar el sistema DOS y finalmente poner el Linux con el gestor de arranque que nos permita seleccionar el sistema con el que iniciamos.

En general el propio instalador detectará que tenemos una partición DOS y la añadirá a las opciones de arranque. Si por cualquier motivo no fuera así debemos modificar el fichero */boot/grub/menu.lst* para que incluya lo siguiente³:

```
title      Sistema_DOS
root      (hd0,0)
makeactive
chainloader + 1
```

También es recomendable para evitar problemas con sistemas que contengan varios discos SATA o SCSI realizar el inicio de sistema basado en etiquetas (*label*). La idea consiste en explicitar que se inicie de una partición con un nombre concreto (no siempre podemos conocer si la memoria USB será el primer, segundo, tercer disco). Para realizar este cambio primero asignamos el nombre a la partición dónde hemos instalado el sistema Linux (en este ejemplo asignamos el nombre “*USB-LINUX*” a la partición */dev/sda2*):

```
# tune2fs -L USB-LINUX /dev/sda2
```

A continuación sustituimos en el fichero *menu.lst* del Grub la línea del kernel dónde se indica la partición del sistema raíz (*root*). Y finalmente actualizamos el Grub como se detalla en el punto 3.4.1.

```
title Linux-Original
root (hd0,1)
kernel /boot/vmlinuz-2.6.26-2 root=dev/sda2 ro
initrd /boot/initrd.img-2.6.26-2

Pasa a ser:

title Linux-Modificado
root (hd0,1)
kernel /boot/vmlinuz-2.6.26-2 root=LABEL=USB-LINUX ro
initrd /boot/initrd.img-2.6.26-2
```

Podemos extender este sistema de identificación de particiones al resto del sistema. De esta forma únicamente debemos modificar el */etc/fstab* de forma que monte las particiones por su nombre (*label*).

³ Podemos ver una explicación con más detalle en el punto 3.4.1 de este documento.

2.5. Copia de la imagen creada a la memoria USB

Una vez que hemos finalizadas las instalaciones en el disco duro virtual, únicamente debemos volcar la imagen en nuestra memoria Flash. Nos debemos asegurar de tener asociada la unidad USB y realizar el comando complementario al inicial (punto 2.1). En nuestro ejemplo:

```
# cat /dev/hda > /dev/sda
```

3. Método 2: Instalación tradicional directamente en la memoria Flash.

En esta segunda parte describiremos los pasos necesarios para instalar uno o varios sistemas operativos directamente en la memoria USB. Este proceso es el más pesado ya que debemos realizar muchos pasos concretos para la personalización del sistema y conseguir que cada una de las particiones se pueda iniciar (bootable).

No es tan flexible como el anterior ya que dependemos del sistema que tenemos instalado en nuestro PC. Reiniciar el ordenador para comprobar si nuestra Flash funciona es otro engorro derivado de este sistema. En contrapartida tenemos un control y una personalización del sistema prácticamente sin límites.

3.1. Creación de las particiones en la memoria Flash.

Este apartado está pensado para realizar la instalación desde cero en una memoria Flash partiendo de un ordenador que ya tenga instalado cualquier versión de Debian (aunque no debería haber problemas con otras distribuciones).

Una vez conectada la memoria Flash en el ordenador, procederemos a comprobar que el sistema la detecta. Por ejemplo mediante el comando “*dmesg*” o examinando los logs del sistema (*/var/log/messages*).

```
kernel: usb 5-6: new high speed USB device using ehci_hcd and address 4
Oct 1 08:28:12 pct1 kernel: usb 5-6: configuration #1 chosen from 1 choice
Oct 1 08:28:12 pct1 kernel: scsi3 : SCSI emulation for USB Mass Storage devices
Oct 1 08:28:17 pct1 kernel: Vendor: Kingston Model: DataTraveler II Rev: 1.13
Oct 1 08:28:17 pct1 kernel: Type: Direct-Access ANSI SCSI revision: 00
Oct 1 08:28:17 pct1 kernel: SCSI device sdc: 2014208 512-byte hdwr sectors (1031 MB)
Oct 1 08:28:17 pct1 kernel: sdc: Write Protect is off
Oct 1 08:28:17 pct1 kernel: SCSI device sdc: 2014208 512-byte hdwr sectors (1031 MB)
Oct 1 08:28:17 pct1 kernel: sdc: Write Protect is off
Oct 1 08:28:17 pct1 kernel: sdc: sdc1
Oct 1 08:28:17 pct1 kernel: sd 3:0:0:0: Attached scsi removable disk sdb
```

Una vez reconocida la memoria Flash por el sistema podemos examinar las particiones existentes en el dispositivo y reparticionar la memoria USB si fuera necesario. En el ejemplo siguiente podemos observar cómo el sistema identifica la memoria Flash USB como el dispositivo */dev/sdc* y tiene una única partición DOS (*sdc1*) con el sistema FAT32.

```
# fdisk -l
```

```
Disco /dev/sdc: 1031 MB, 1031274496 bytes
16 cabezas, 32 sectores/pista, 3934 cilindros
Unidades = cilindros de 512 * 512 = 262144 bytes
```

Disposit.	Inicio	Comienzo	Fin	Bloques	Id	Sistema
/dev/sdc1	*	1	3934	1007088	b	W95 FAT32

A continuación procedemos a borrar TODA la memoria Flash y por tanto sus contenidos. Esto es importante porque muchas memorias vienen formateadas con opciones que imposibilitan usarlas como sistemas de inicio (bootable system). **Es importante asegurarse que borramos la memoria USB y no un disco del sistema. ¡Esta operación elimina todos los datos existentes!**

```
# dd if=/dev/zero of=/dev/sdc bs=512 count=1
```

```
1+0 records in
1+0 records out
512 bytes (512 B) copied, 0,00961093 seconds, 53,3 kB/s
```

A continuación creamos un nuevo MBR (Master Boot Record) en la memoria Flash mediante las utilidades “ms-sys”. Cabe destacar que en los sistemas nuevos, por ejemplo Debian Lenny, estas utilidades se han eliminado por problemas de patentes con Microsoft [www20]. Si al intentar instalarlas obtenemos un error de “paquete desconocido” o similar, deberemos bajarlas de su web [www19] y compilarlas manualmente. En otros casos o sistemas antiguos simplemente deberemos ejecutar:

```
# apt-get install ms-sys dosfstools
```

Una vez instaladas o compiladas las utilidades procedemos con la instalación del “boot record” que nos permitirá convertir una partición en iniciable o *bootable*.

```
# ms-sys --mbrdos /dev/sdc
```

```
DOS/Windows NT master boot record successfully written to /dev/sdc
```

Finalmente procedemos a crear las particiones con el tamaño deseado mediante el comando “*fdisk*” o cualquier otra utilidad que permita reparticionar el sistema. Es importante señalar que en el caso de sistemas duales, la partición de MS-DOS ha de ser la primera que se encuentre en la memoria. Nosotros utilizaremos para MS-DOS el sistema FAT32 (tipo 0x0b) y para el sistema Linux el 0x83.

```
# fdisk -l
```

```
Disco /dev/sdc: 1031 MB, 1031274496 bytes
16 cabezas, 32 sectores/pista, 3934 cilindros
Unidades = cilindros de 512 * 512 = 262144 bytes
```

Disposit.	Inicio	Comienzo	Fin	Bloques	Id	Sistema
/dev/sdc1	*	1	1000	407088	b	W95 FAT32
/dev/sdc2		1001	3934	954566	83	Linux

En el caso de utilizar dos particiones, la de MS-DOS ha de ser la de inicio. Este flag se activa mediante la opción “a” del *fdisk*.

3.2 Formateo del sistema de ficheros MS-DOS y Linux

Una vez creada la partición, o particiones, en la memoria Flash nos disponemos a formatear la partición Linux usando un sistema preferiblemente con journaling como ext3 (todo y que seria posible usar cualquier otro sistema de ficheros).

```
# mkfs.ext3 -j -L USB /dev/sdc2
```

```
Etiqueta del sistema de ficheros=USB
Tipo de SO: Linux
Tamaño del bloque=1024 (bitácora=0)
Tamaño del fragmento=1024 (bitácora=0)
25064 nodos i, 100160 bloques
5008 bloques (5.00%) reservados para el súper usuario
Primer bloque de datos=1
Maximum filesystem blocks=67371008
13 bloque de grupos
8192 bloques por grupo, 8192 fragmentos por grupo
1928 nodos i por grupo
Respaldo del súper bloque guardado en los bloques:
    8193, 24577, 40961, 57345, 73729
```

```
Mientras se escribían las tablas de nodos i: terminado
```

```
Creando el fichero de transacciones (4096 bloques): hecho
Escribiendo superbloques y la información contable del sistema de
ficheros: hecho
```

```
Este sistema de ficheros se revisará automáticamente cada 37 meses o
180 días, lo que suceda primero. Utilice tune2fs -c o -i para cambiarlo.
```

En nuestro caso, además de formatear el sistema con *journaling* lo etiquetamos (-L label) con el nombre “USB”. Esto nos permitirá más adelante montar el sistema independientemente del dispositivo. Podemos obtener información del sistema de ficheros creado mediante la utilidad *tune2fs* (por ejemplo podemos ejecutar: *tune2fs -l /dev/sdc2*).

Para la partición MS-DOS de tipo FAT32 realizaremos también el formateo del sistema de ficheros mediante la utilidad “*mkfs.vfat*”. En caso de ser necesario se podría realizar el mismo procedimiento con FAT16 modificando el flag “-F”. La elección del sistema FAT32 [www15] es debida a que FAT16 tiene un tamaño máximo de partición de 2Gbytes y su tamaño de cluster (unidad básica de direccionamiento en el sistema de ficheros) crece excesivamente con el tamaño de partición. FAT32 es más equilibrado en estos aspectos.

```
# mkfs.vfat -F 32 -n USB-FAT32 /dev/sdc1
```

```
mkfs.vfat 2.11 (12 Mar 2005)
```

3.3. Instalación base del sistema operativo MS-DOS

Una vez inicializado el sistema de ficheros FAT32 únicamente debemos copiar los archivos “*io.sys*”, “*msdos.sys*” y “*command.com*” para obtener un sistema básico de arranque. Estos ficheros

pueden obtenerse desde el propio sistema Windows 95/98 o formateando un diskette y añadiendo la opción de disco de sistema o de algunas otras fuentes como [www11][www12] [www13].

```
# mount /dev/sdc1 /mnt
# cp io.sys /mnt
# cp msdos.sys /mnt
# cp command.com /mnt
# umount /mnt
```

En el caso de querer copiar el Hiren's BootCD, y mantener sus ficheros originales deberemos obtener la imagen iso original y extraer los ficheros de inicio de sistema junto a los menús (*config.sys*, *autoexec.bat*...). Esto lo podemos conseguir fácilmente con la utilidad “*extract-boot-files.zip*” [www14] que dada una imagen del Hiren's BootCD extrae los ficheros necesarios en “C:\USB”.

Cabe destacar que este procedimiento sólo funciona con versiones anteriores a la 9.8 debido a que en esta versión se realizó un cambio del sistema de arranque. En caso de no funcionar este sistema únicamente debemos copiar los ficheros de configuración (*config.sys*, *autoexec.bat*...) a mano.

(desde Windows o MS-DOS)

```
C:\> Get_Boot_Files.cmd
```

```
All Done
All Files copied in C:\USB
```

```
Presione una tecla para continuar . . .
```

A continuación también copiaremos los archivos que se incluyen en la imagen iso del CD-ROM y todo el directorio “*bootcd*”. Esto ya podemos realizarlo directamente en Linux porque nos permite montar imágenes iso.

```
# mount -o loop Hiren'sBootcd.iso /mnt
```

Finalmente debemos marcar la memoria Flash y la partición de MS-DOS como activa (bootable) mediante la utilidad “*ms-sys*”.

```
# ms-sys -w /dev/sdc
```

```
DOS/Windows NT master boot record successfully written to /dev/sdc
```

```
# ms-sys -w /dev/sdc1
```

```
FAT32 DOS boot record successfully written to /dev/sdc1
```

```
# ms-sys -p /dev/sdc1
```

```
Start sector (nr of hidden sectors) successfully written to /dev/sdc1
Physical disk drive id 0x80 (C:) successfully written to /dev/sdc1
```

Finalmente ya podemos reiniciar el ordenador y comprobar cómo se inicializa nuestro sistema basado en MS-DOS desde la memoria USB. Es importante destacar que algunas placas bases permiten seleccionar el sistema de inicio mediante alguna tecla (F12) mientras comprueba la BIOS mientras que en otros sistemas han de configurarse desde dentro de las opciones de inicio de sistema de la BIOS (*boot sequence*).

3.4. Instalación base del sistema operativo Linux Debian

Una vez preparada la memoria Flash, procedemos a montar la partición en el “/mnt” y ejecutar el instalador de Debian. Este instalador se encuentra disponible en el paquete “bootstrap” [www3]. Cabe destacar que podemos elegir la versión y arquitectura de Debian que deseemos instalar. De esta forma si deseásemos instalar una versión “etch” simplemente deberíamos ejecutar el comando: **debootstrap --arch i386 etch /mnt**

```
# mount /dev/sdc2 /mnt
# debootstrap --arch i386 lenny /mnt
```

Tras obtener los paquetes mínimos necesarios que se instalan, montaremos en la memoria USB el /dev y el /proc para continuar la instalación. Los montaremos con la opción *--bind* para que utilicen los mismos que ya existen en el sistema actual.

```
# mount --bind /dev/ /mnt/dev/
# mount --bind /proc /mnt/proc

# chroot /mnt /bin/sh -l
```

Una vez preparado el sistema base en la memoria Flash, accedemos a él para personalizarlo creando los ficheros básicos de configuración del sistema. Usamos como identificador la etiqueta “USB” para aislarnos de los dispositivos físicos. De esta forma nos independizamos de si es /dev/sda1, /dev/sdc2...

```
#
# Fichero /etc/fstab
#
LABEL=USB / ext3 defaults,errors=remount-ro,noatime 0 1
none /proc proc defaults 0 0
```

En caso de ir escasos de espacio en la partición de la memoria USB es recomendable ir borrando los paquetes .deb una vez instalados mediante el comando “*apt-get clean*”. Para continuar con la personalización del sistema creamos los ficheros de configuración de red */etc/hosts*, */etc/resolv.conf* y */etc/network/interfaces*.

```
#
# Fichero /etc/hosts
#
127.0.0.1 localhost.localdomain localhost
# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
```

En la bibliografía adjunta [www5][www6][www7] podemos obtener un listado con los diferentes servidores DNS de los distintos operadores o ISP.

```
# Fichero /etc/resolv.conf
#
search mi_dominio.es
nameserver X.Y.Z.T
```

```
#  
# Fichero /etc/network/interfaces  
#  
  
# The loopback network interface  
auto lo  
iface lo inet loopback  
  
# The primary network interface  
auto eth0  
iface eth0 inet dhcp
```

Finalmente creamos el fichero `”/etc/apt/sources.list”` con las fuentes de los paquetes Debian. Aquí se contemplan todas las fuentes oficiales disponibles, pero se puede personalizar esta parte dependiendo de las necesidades de cada uno.

```
# Example of Debian resources for Lenny  
  
deb ftp://ftp.debian.org/debian/ lenny main contrib non-free  
deb http://security.debian.org/ lenny/updates main contrib non-free  
deb http://volatile.debian.org/debian-volatile lenny/volatile main contrib non-free
```

En el caso de la versión *Etch* de Linux, para evitar el aviso de que falta una clave PGP al ejecutar el comando `“apt-get update”`. Debemos ejecutar la siguiente secuencia de comandos que nos permiten obtener la clave PGP.

```
# gpg --homedir /root --keyserver wwwkeys.eu.pgp.net --recv-keys F1D53D8C4F368D5D  
# gpg --homedir /root --armor --export F1D53D8C4F368D5D | apt-key add -
```

3.4.1. Actualización del sistema Debian ya instalado

A continuación actualizamos el sistema con las fuentes existentes y añadimos los componentes básicos que permitirán iniciar el sistema (booting) desde la memoria USB.

```
# apt-get update  
# apt-get install initramfs-tools
```

Una vez instaladas las utilidades *initramfs* necesitamos configurar el fichero *initramfs.conf* que será necesario para instalar el kernel y añadir los módulos correctamente al fichero *initrd.img*.

```
#/etc/initramfs-tools/initramfs.conf  
# Configuration file for mkinitramfs(8). See initramfs.conf(5).  
MODULES=most  
BUSYBOX=y  
BOOT=local  
DEVICE=eth0  
NFSROOT=auto
```

Ahora ya podemos realizar la instalación del gestor de arranque. Nosotros instalaremos el *GRUB* que es un gestor bastante flexible, todo y que en principio es posible instalar el *LILO*. En los nuevos sistemas se nos da la opción de instalar GRUB2 pero debido a los problemas que he encontrado recomiendo instalar GRUB versión 1 o GRUB *legacy*.

apt-get install grub mdadm

Con el gestor de arranque LILO he observado que en algunos sistemas antiguos con USB 1.1 las memorias Flash no se inicializan bien. De todas formas hay muchos testimonios vía Google de gente que los usa sin problemas.

La instalación del paquete MDADM (raid) no es necesaria, pero recomiendo su instalación por si nos encontramos con algún sistema que lo soporte. Finalmente salimos de nuestra jaula “chroot” para realizar la instalación del gestor de arranque en el dispositivo de la memoria Flash (/dev/sdc) y comprobar la numeración de los dispositivos.

```
# exit
# grub-install --recheck --root-directory=/mnt /dev/sdc

Probing devices to guess BIOS drives. This may take a long time.
Installation finished. No error reported.
This is the contents of the device map /mnt/boot/grub/device.map.
Check if this is correct or not. If any of the lines is incorrect,
fix it and re-run the script `grub-install'.
(fd0) /dev/fd0
(hd0) /dev/sda
(hd1) /dev/sdc
```

Una vez conocida la equivalencia entre los dispositivos (/dev/sdXX) y la nomenclatura que utiliza el *GRUB* (hd0, hd1...) podemos volver a nuestro sistema con la jaula *chroot*.

```
# chroot /mnt /bin/sh -l
```

Finalmente crearemos el fichero básico para el inicio del gestor de arranque seleccionado que se denomina */boot/grub/menu.lst*.

```
# /boot/grub/menu.lst mínimo

# default num
default      0

# timeout sec
timeout      5

# pretty colours
color green/black black/green

### BEGIN AUTOMAGIC KERNELS LIST

## ## Start Default Options ##

# kopt=root=/dev/mapper/rootfs ro\n")
# groot=(hd0,0)
# alternative=true
# lockalternative=false
# defoptions=
# lockold=false
# howmany=all
# memtest86=true
# updatedefaultentry=false

## ## End Default Options ##

# Para iniciar la primera partición (MSDOS o Hiren'sBootCD)
```



```
Title      MS-DOS
root       (hd0,0)
chainloader +1
```

```
### END DEBIAN AUTOMAGIC KERNELS LIST
```

También recomendamos la instalación de los paquetes locales que contienen la configuración necesaria para la configuración del idioma por defecto y el teclado. El paquete “*localepurge*” es importante ya que una vez seleccionado el idioma por defecto, eliminará todos los otros permitiéndonos ganar algo de espacio en la memoria USB.

```
# apt-get install locales localepurge kbd console-common unicode-data
# apt-get clean
```

Si fuera necesario reconfigurar el idioma (*locale*) seleccionado o deseamos seleccionar el idioma del teclado de nuestro sistema, podemos ejecutar los comandos:

```
# dpkg-reconfigure locales      y      # dpkg-reconfigure console-data
```

3.4.2. Instalación del Kernel en el sistema preinstalado

Antes de proceder a la instalación del kernel, debemos crear el fichero de configuración pertinente */etc/kernel-img.conf*.

```
do_symlinks = yes
relative_links = yes
do_bootloader = no
do_bootfloppy = no
do_initrd = yes
link_in_boot = yes
postinst_hook = /usr/sbin/update-grub
postrm_hook = /usr/sbin/update-grub
```

A continuación instalaremos el kernel deseado. En este ejemplo instalaremos el kernel más genérico que hemos encontrado en Debian. En cada caso elige el que más te convenga.

```
# apt-get install linux-image-2.6-486
```

Para finalizar la actualización del sistema Debian en la memoria Flash y comprobar que el gestor de arranque se encuentra perfectamente instalado, ejecutaremos el *GRUB*.

```
# grub

GNU GRUB version 0.97 (640K lower / 3072K upper memory)

[ Minimal BASH-like line editing is supported.  For
the first word, TAB lists possible command
completions.  Anywhere else TAB lists the possible
completions of a device/filename. ]

grub> root (hd1,1)
root (hd1,1)
Filesystem type is ext2fs, partition type 0x83
```

```
grub> setup (hd1)
setup (hd1)

Checking if "/boot/grub/stage1" exists... yes
Checking if "/boot/grub/stage2" exists... yes
Checking if "/boot/grub/e2fs_stage1_5" exists... yes
Running "embed /boot/grub/e2fs_stage1_5 (hd1)"...
Running "embed /boot/grub/e2fs_stage1_5 (hd1,1)"...
Running "install /boot/grub/stage1 (hd1) /boot/grub/stage2
/boot/grub/menu.lst "... succeeded
Done.

grub> quit
```

Antes de probar que todo funciona borraremos el fichero de mapeo de dispositivos para que lo busque cada vez que se inicie. Esto es recomendable ya que este fichero creado contiene el mapeo correspondiente al sistema dónde hemos creado la memoria Flash, que probablemente no sea el mismo dónde lo probemos. También podemos aprovechar y eliminar los paquetes .deb ya instalados en el sistema.

```
# rm -f /boot/grub/device.map
# apt-get clean
```

3.5. Ampliaciones del sistema Debian Linux instalado en una memoria Flash USB

Una vez configurado el sistema básico Debian podemos añadir diferentes servicios y programas (mediante el *apt-get/dselect* de toda la vida o el *aptitude* por ejemplo) para personalizar nuestro nuevo sistema USB que nos acompañará a todos los lugares. Obviamente siempre hay que tener en cuenta el tamaño de la memoria Flash, dejando al menos 10/15MBytes libres para poder movernos mínimamente por el sistema.

Ejemplos de ampliaciones interesantes a nuestro sistema podrían ser:

- **Crear usuarios:** Por defecto no hay usuarios creados en la instalación (excepto *root*). Una buena práctica consistiría en deshabilitar el usuario root (¡que por defecto NO tiene password!) y crear al menos un usuario normal con el que trabajar mediante sudo [www29].
- **Instalar el servidor de SSH:** De esta forma permitiríamos conexiones desde el exterior lo que nos permitiría de subir y bajar ficheros mediante SecureFTP por ejemplo.
- **Instalar el sistema gráfico (X-Windows):** Consume algunos recursos del sistema, sobre todo espacio de disco, pero la verdad es que es útil. Un gestor de ventanas ligero y funcional como “*blackbox*” pueden ser un muy buen candidato.
- **Clonar el Linux de tu ordenador en la memoria Flash:** Como es de suponer podemos no instalar un sistema desde cero y simplemente copiar todo nuestro sistema Debian (o cualquier otra distribución de Linux) en la memoria USB siempre y cuando tengamos espacio suficiente en la Flash.

Simplemente deberemos hacer un “tar” de todo nuestro sistema, copiarlo en la partición Linux de nuestra memoria USB y ejecutar el Grub para instalar un gestor de arranque. Por ejemplo podemos realizar alguna cosa del estilo siguiente:

```
# mount /dev/sdc2 /mnt
# tar --exclude=/mnt -cf - / | tar -C /mnt --xvf -
```

Y a continuación ejecutar la instalación y personalización del Grub.

- **Montar todo el sistema en memoria (RAM):** Una opción interesante teniendo en cuenta que los equipos hoy en día vienen con más de un Gigabyte de RAM, puede ser el de montar parte del sistema en memoria. Esta opción nos permite tratar a la memoria Flash como si fuera un CD-Rom, ya que únicamente leeremos el sistema y NO se escribirán los cambios que realicemos en los directorios montados en RAM mediante el *tmpfs*.

La ventaja es que siempre que iniciemos el sistema con nuestra memoria USB tendremos el mismo sistema ya que los cambios en los ficheros/directorios temporales en RAM se pierden al reiniciar el sistema. La desventaja es que no podemos consultar los logs históricamente puesto que no quedan guardados en la memoria Flash. Por ejemplo creamos este */etc/fstab*.

```
#/etc/fstab: static file system information.
#
LABEL=USB / ext3 defaults,errors=remount-ro,noatime 0 1
none /proc proc defaults 0 0
tmpfs /tmp tmpfs defaults,noatime 0 0
tmpfs /var/lock tmpfs defaults,noatime 0 0
tmpfs /var/log tmpfs defaults,noatime 0 0
tmpfs /var/run tmpfs defaults,noatime 0 0
tmpfs /var/tmp tmpfs defaults,noatime 0 0
```

- **Montar FS cifrados:** En caso de ser ligeramente paranoico o simplemente querer añadir seguridad al sistema, podemos crear un sistema de ficheros cifrado [www1]. En este caso simplemente hay que crear dos particiones para Linux en vez de una a la hora de instalar el sistema o mover los archivos una vez finalizada. El usuario root tiene sus datos en */root* con lo que si queremos que también tenga un sistema cifrado, deberemos moverlo a esta partición.

Posibilidad A: Se destinará la primera partición Linux al */boot*. Esta partición ha de estar sin cifrar, ya que al iniciar el sistema no hay manera de leer y descifrar el sistema de ficheros.

La segunda partición contendrá el resto del sistema (*/home, /bin, /usr, /var...*) que sí estará cifrada. Obviamente el kernel que utilicemos ha de tener soporte para sistemas cifrados.

Posibilidad B: Se destina la primera partición (sin cifrar) a todo el sistema Linux excepto el */home*. En la segunda partición se incluye el */home* y por tanto todos los datos de usuario quedarán cifrados.

Para realizar este procedimiento se puede realizar la instalación normal del Linux en la memoria USB descrita en este documento y una vez finalizada mover el */home* a la segunda partición.

4. Problemas

Si por cualquier extraña casualidad siguiendo las indicaciones de este documento y con algo de tu parte todo te funciona correctamente, puedes enviarme un email. Como decía aquel “*get new mail feeds my ego*” :)

Para todos los demás, aquí recojo una selección de los problemas más comunes que podéis encontrar:

1. No puedo iniciar mi ordenador desde la memoria USB

Si tu ordenador te muestra la lista de dispositivos para seleccionar el dispositivo de inicio, asegúrate que seleccionas correctamente la Flash. Muchos portátiles ven la disquetera y el CD-Roms como dispositivos USB.

Si no llegas a este punto, pulsando generalmente F12 justo al iniciar el ordenador o portátil mientras se comprueba la BIOS, asegúrate que tu sistema permite iniciar dispositivos USB. Consulta el manual de la placa base, mira las opciones existentes en la BIOS o prueba a actualizarla.

2. La tarjeta de red va creciendo “eth0”, “eth1”, “eth2”...

Con el sistema de ficheros “*udev*” [www16] se tiende a realizar una cache de dispositivos en el directorio `/etc/udev/rules.d`. De esta manera al cambiar el sistema de ordenador los dispositivos antiguos tienen ya ocupados los nombres (eth0, eth1...) y el sistema utiliza los siguientes que son los que tiene libres.

El fichero `/etc/udev/rules.d/z25_persistent-net.rules` es el encargado de mantener la cache de dispositivos de red.

```
# This file was automatically generated by the /lib/udev/write_net_rules
# program, probably run by the persistent-net-generator.rules rules file.
#
# You can modify it, as long as you keep each rule on a single line.
# MAC addresses must be written in lowercase.

# PCI device 0x10ec:0x8139 (8139too)
SUBSYSTEM=="net", DRIVERS=="?*", ATTRS{address}=="00:50:fc:2b:a0:XX NAME="eth0"

# PCI device 0x1106:0x3065 (via-rhine)
SUBSYSTEM=="net", DRIVERS=="?*", ATTRS{address}=="00:0c:6e:c2:f7:XX NAME="eth1"
```

De esta forma al borrar este fichero cada vez que se inicie el sistema nos asegurarnos de que las tarjetas de red empiecen en el dispositivo *eth0*.

Por ejemplo podemos añadir las siguientes líneas al fichero `/etc/init.d/syslogd`

```
echo "" > /etc/udev/rules.d/z25_persistent-net.rules
echo "" > /etc/udev/rules.d/z25_persistent-cd.rules
```

3. No aparece en pantalla ni el mensaje “GRUB Loading”

Si el ordenador o portátil sí permite seleccionar la memoria Flash como dispositivo de inicio pero no aparece ningún mensaje en la pantalla puede pasar:

- Que no exista ninguna partición activa. Recuerda que todo y no ser estrictamente necesario si el gestor de arranque está bien instalado, debería existir una partición activa en la memoria USB. Si tienes una partición MS-DOS esta debe ser la primera de la memoria Flash y debe estar activa.
- Que no hayas ejecutado correctamente el “*grub-install*” o el “*grub*” o que lo hayas instalado en el disco o partición equivocado.

4. No puedo iniciar el sistema Linux

Si el problema viene una vez cargado el kernel al iniciarse el Linux, probablemente tengas problemas con el “*root filesystem*”. Recuerda que al formatear el sistema es necesario incluir la opción `-L USB` para que pueda encontrar la partición adecuada. En el caso de ext3 puedes comprobar el sistema para ver la etiqueta (*Label*) mediante el siguiente comando:

```
# tune2fs -l /dev/sdc2
```

Si has compilado un kernel nuevo asegúrate de poner la imagen *bzImage*, el *initrd.img* en el */boot* y que los nombres del *menu.lst* se corresponden con los que has instalado. También se ha de tener en cuenta que los módulos correspondientes han de encontrarse en el directorio */lib/modules*.

5. No puedo iniciar el sistema MS-DOS

Desgraciadamente este es el problema más común y el de más difícil solución. Antes de nada debemos intentar averiguar si el “problema” es que el ordenador o portátil concreto dónde lo estás probando no permite iniciar sistemas MS-DOS desde una memoria Flash o simplemente es que alguno de los pasos anteriores no se ha realizado correctamente.

Si disponemos de otro ordenador, lo más sencillo es comprobar si se inicia en otros ordenadores. Si no es posible comprobarlo en otro sitio, la clave puede ser comprobar si al iniciarse la memoria USB muestra el menú de Grub.

Si aparece el menú del Grub, casi seguro que al ordenador no le gusta el sistema DOS en memorias Flash, ya que el gestor de arranque sí lo ha leído, y por tanto la memoria la reconoce bien. A fin de cuentas la creación de un sistema MS-DOS únicamente implica dos pasos sencillos con el **ms-sys** y copiar como mínimo los ficheros “*msdos.sys*”, “*io.sys*” y “*command.com*”. Para comprobar que has ejecutado correctamente los pasos anteriores, puedes mirar los sectores de inicio de la memoria USB ejecutando los siguientes comandos:

```
# ms-sys /dev/sdc    y    # ms-sys /dev/sdc1
```

De todas maneras se podría probar a copiar otra versión de estos ficheros, por si estuvieran dañados o fueran de una versión muy antigua o que causara problemas. También se puede probar variando el tamaño de la partición MS-DOS, ya que puede ayudar en algunas circunstancias.

6. No puedo quitar el gestor de inicio GRUB de la memoria Flash

Hay varias maneras de eliminarlo ya que simplemente debemos instalar un nuevo MBR. Una forma sencilla de realizarlo es mediante el programa fdisk de MS-DOS o Windows con la opción */mbr* o desde Linux utilizando el siguiente comando:

```
# ms-sys --mbrdos /dev/sdc
```

5. Copyright

Este documento se distribuye bajo la licencia Creative Commons 2.5 que permite la difusión libre de este documento debiendo siempre respetar y citar en los créditos a su autor y prohibiendo el uso comercial sin expresa autorización del autor.

<http://creativecommons.org/licenses/by-nc/2.5/es/>



Bibliografía

- [www1] <http://feraga.com/node/30>
- [www2] <http://www.vicente-navarro.com/blog/index.php/2007/06/29/arrancar-ms-dos-desde-una-memoria-usb/>
- [www3] <http://packages.debian.org>
- [www4] <http://ms-sys.sourceforge.net/>
- [www5] <http://www.adslayuda.com/modulo-FSDns.html>
- [www6] <http://www.bandaancha.st/toolsdns.php>
- [www7] <http://www.adslzone.net/dns.html>
- [www8] <http://es.wikipedia.org/wiki/MS-DOS>
- [www9] <http://es.wikipedia.org/wiki/PC-DOS>
- [www10] <http://www.hiren.info/pages/bootcd>
- [www11] <http://www.bootdisk.com/>
- [www12] http://www.allbootdisks.com/disk_contents/dos.html
- [www13] <http://freepctech.com/pc/002/files010.shtml>
- [www14] <http://www.hiren.info/download/freeware/extract-boot-files.zip>
- [www15] http://es.wikipedia.org/wiki/Tabla_de_Asignaci%C3%B3n_de_Archivos
- [www16] <http://www.kernel.org/pub/linux/utils/kernel/hotplug/udev.html>
- [www18] <http://www.virtualbox.org/>
- [www19] <http://ms-sys.sourceforge.net/>
- [www20] <http://packages.qa.debian.org/m/ms-sys/news/20080405T223920Z.html>
- [www21] <http://cdimage.debian.org/debian-cd/5.0.1/i386/iso-cd/debian-501-i386-netinst.iso>
- [www22] <http://www.livedcdlist.com/>
- [www23] <http://en.wikipedia.org/wiki/Fat16>
- [www24] <http://www.nu2.nu/pebuilder/>
- [www25] <http://asuse3.blogspot.com/2008/01/ejecutar-windows-xp-en-la-ram-vdeo.html>
- [www26] <http://gocoding.com/page.php?al=pitousb>
- [www27] http://www.bootcd.us/BartPE_Plugins_Repository.php
- [www28] <http://www.microsoft.com/windowseembedded/en-us/products/wexpe/default.mspx>
- [www29] <http://en.wikipedia.org/wiki/Sudo>
- [www30] <http://www.microsoft.com/downloads/details.aspx?familyid=dacd1722-256b-48c5-91c1-af6062340efc&displaylang=en>