

# Analizando en profundidad el proyecto METADISTROS. El proceso de Detección de Hardware.

*Versión 0.x.x*

Rafael Martín de Agar Tirado

[rmartin@emergya.info](mailto:rmartin@emergya.info)

(Emergya, S.C.A.)

Fecha: 8. Sep. 2003

---

Copyright © 2003 Rafael Martín de Agar Tirado.

Se otorga permiso para copiar, distribuir y/o modificar este documento bajo los términos de la Licencia de Documentación Libre GNU, versión 1.2 o cualquier otra versión posterior publicada por la Fundación de Software Libre.

Se incluye una copia de esta licencia en la sección que lleva por título "Licencia de Documentación Libre GNU"

# Índice de Contenidos

1.Introducción.....	3
2.El proceso de arranque de un sistema GNU/Linux .....	4
2.1El comienzo del arranque.....	4
2.2El proceso 'init'. Los servicios básicos y el nivel de ejecución.....	4
2.2.1Los servicios básicos.....	4
2.2.2El nivel de ejecución.....	5
2.3Arrancando con 'initrd' .....	5
3.El proceso de arranque en Meta-Distros.....	6
3.1Configuración de ISOLINUX.....	6
3.2El archivo más importante: initrd.gz.....	7
3.2.1Estructura del contenido de initrd para Meta-Distros.....	7
El directorio '/'.....	7
El directorio /META.....	8
El directorio 'modules'.....	8
El directorio 'keymaps'.....	8
El directorio 'dev'.....	9
El directorio 'etc'.....	9
El directorio 'static': BusyBox.....	9
El fichero '/linuxrc': el control del arranque en initrd.....	9
3.3Continuación del arranque: el proceso INIT.....	16
3.3.1Análisis del diagrama de flujo.....	16
3.3.2El proceso de detección de Hardware: hw-detect.sh.....	21
4.El sistema de detección de Hardware en MetaDistros.....	24
4.1Primera fase: initrd.....	24
4.2Segunda fase: proceso init.....	24
4.3Cómo añadir soporte para nuevos dispositivos.....	24
5.Funcionamiento de KUDZU.....	25
5.1Descripción.....	25
5.2Funcionamiento.....	25
5.3Opciones.....	26
5.4Ficheros relacionados.....	27
5.4.1/etc/sysconfig/hwconf.....	27
5.4.2/etc/sysconfig/kudzu.....	27
5.4.3/etc/modutils/kudzu.....	27
5.4.4/etc/sysconfig/network-scripts/ifcfg-*.....	28
6.Bibliografía.....	29
7.GNU Free Documentation License .....	30

# 1.Introducción

El proyecto MetaDistros<sup>1</sup>, uno de los múltiples proyectos puestos en marcha por HispaLinux, tiene como objetivo la creación, de una manera sencilla, de una distribución de GNU/Linux a medida, completamente personalizada para las necesidades de un usuario particular.

Este proyecto, aunque se encuentra aún en una etapa muy inicial, es ya una realidad: varias decenas de distribución basadas en MetaDistros han visto ya la luz.

Sin embargo, hay un punto especialmente débil en todo el proyecto: la falta de documentación técnica. Cuando una persona se adentra en el mundo de MetaDistros con el ánimo de crear una distribución a medida, dispone de algunos manuales y, en particular, de un HOWTO, que explica de manera sencilla y clara cómo puede hacerlo.

Por el contrario, cuando una persona quiere contribuir al desarrollo del proyecto, la cosa cambia, precisamente por la falta de documentación técnica. Éste fue precisamente mi caso particular.

Tenía la idea de avanzar, mejorar y depurar el proceso de detección de hardware de MetaDistros, puesto que me parece una parte crítica del sistema. Pero me encontré con el problema de que no había ningún documento que comentara la manera en que funciona internamente el sistema de MetaDistros.

Así, me tuve que poner a “buscarme la vida”, tratando de encontrar cualquier documento que pudiera aclararme un poco las ideas.

Después de mucho leer y leer y leer, llegué a la conclusión de que esto no debía repetirse para cada persona que quisiera contribuir en este proyecto, así que trasladé mis esfuerzos al desarrollo de este documento.

El objetivo que persigue es proporcionar una visión general del funcionamiento interno de MetaDistros, haciendo hincapié en cómo se produce el arranque del sistema y la detección de hardware. Asimismo, se pretende dar una visión de las partes que componen una MetaDistribución y cuál es la función que aporta cada una de ellas.

Este documento se ha estructurado de la siguiente forma:

- Una primera parte, en la que se describe de manera genérica cómo se produce el proceso de arranque en un sistema GNU/Linux.
- Una segunda parte, en la que se comenta las modificaciones que ha sido necesario llevar a cabo para permitir el arranque de una distribución generada a partir de MetaDistros (y, por tanto, con ciertas peculiaridades que la distinguen de una distribución “normal”).
- La última parte se centra en el proceso de detección de hardware en sí mismo, haciendo un resumen de lo que ya se ha comentado en los apartados anteriores.

---

<sup>1</sup> Más info: <http://metadistros.hispalinux.es>

## 2.El proceso de arranque de un sistema GNU/Linux <sup>2</sup>

### 2.1El comienzo del arranque.

Todos los PCs comienzan el proceso de arranque ejecutando código en ROM (específicamente, la BIOS) para cargar la información del sector 0 y cilindro 0 del disco de arranque. El disco de arranque suele estar configurado normalmente como la primera disquetera. La BIOS intenta ejecutar entonces este sector. En la mayoría de discos arrancables, el sector 0, cilindro 0 contiene una de las dos posibilidades siguientes:

- Código de un gestor de arranque como LILO, que localiza el núcleo, lo carga y lo ejecuta para comenzar el proceso de arranque propiamente dicho.
- El comienzo del núcleo de un sistema operativo, como Linux.

Si un núcleo de Linux se ha copiado a un disquete (mediante una copia *raw*), el primer sector del disquete será el primer sector del núcleo de Linux. Este primer sector continuará el proceso de arranque cargando el resto del núcleo desde el disco de arranque.

Cuando el núcleo se ha cargado completamente, inicializa los *drivers* de los dispositivos y sus estructuras de datos internas. Una vez que ha sido inicializado completamente, consulta una posición especial en su imagen llamada *ramdisk word*. Esta palabra contiene la información de dónde y cómo encontrar el sistema de ficheros raíz. Un sistema de ficheros raíz es simplemente un sistema de ficheros que será montado en el directorio raíz “/”. Hay que decirle al núcleo dónde debe buscar el sistema de ficheros raíz; si no puede encontrar una imagen que se pueda cargar, se para el proceso de arranque.

En algunas situaciones, normalmente cuando se arranca desde un disquete, el sistema de ficheros raíz se carga en memoria, que no es más que memoria RAM a la que se accede como si fuera un disco. La RAM es varios órdenes de magnitud más rápida que un disquete, de manera que se gana en velocidad. Además, el núcleo puede cargar un sistema de ficheros comprimido desde el disquete y descomprimirlo en el disco de memoria, permitiendo así introducir muchos más ficheros en un solo disquete de arranque.

Cuando el sistema de ficheros raíz se ha cargado y montado, se muestra un mensaje similar a éste:

```
VFS: Mounted root (ext2 filesystem) readonly.
```

### 2.2El proceso 'init'. Los servicios básicos y el nivel de ejecución.

#### 2.2.1Los servicios básicos.

Una vez que el sistema ha cargado adecuadamente el sistema de ficheros raíz, se procede a ejecutar el programa “init” (en /bin o /sbin). “init” lee su fichero de configuración (/etc/inittab), busca una línea que contiene “sysinit”, y ejecuta el script asociado. El script que se ejecuta puede variar de una distribución de GNU/Linux a otra. En sistemas Debian, esta línea es similar a:

```
si::sysinit:/etc/init.d/rcS
```

---

<sup>2</sup> Basado en 'The Linux Bootdisk HOWTO'

El script que pasa a ejecutarse (/etc/init.d/rcS) procede a ejecutar, de manera modular, los diferentes scripts que aparecen en /etc/rcS.d/. Éstos se encargan de configurar el sistema de manera básica, iniciando los servicios básicos del sistema, como la ejecución de “fsck” para comprobar el sistema de ficheros raíz, la carga de los módulos del núcleo necesarios, la inicialización del “swap”, de la red, montando los discos especificados en /etc/fstab, etc.

## 2.2.2 El nivel de ejecución.

Cuando el script de sysinit finaliza su ejecución, se devuelve el control al proceso “init”, que entonces procede a entrar en el nivel de ejecución (o *runlevel*) por defecto, especificado en /etc/inittab por la palabra “initdefault”. Será algo parecido a la siguiente línea:

```
id:2:initdefault:
```

El nivel de ejecución elegido incluirá un conjunto de scripts, ubicados en /etc/rcX.d (donde X es el número correspondiente al nivel de ejecución), que procederán a ejecutarse. Por último, en el fichero “/etc/inittab” se especifica un programa como “getty”, que se encarga de manejar las comunicaciones desde la consola y los ttys (terminales remotos). Es este programa el responsable de mostrar el conocido mensaje de “login:”. Se le pasará posteriormente el control al programa “login”, que se encargará de validar la entrada del usuario al sistema.

## 2.3 Arrancando con 'initrd'<sup>3</sup>

El fichero especial /dev/initrd es un dispositivo de bloques de sólo lectura. Este dispositivo es un disco RAM que es inicializado por el gestor de arranque antes de que se arranque el núcleo. Entonces, el núcleo puede utilizar el contenido de /dev/initrd para realizar un arranque en dos fases.

En la primera fase, el núcleo comienza su ejecución y monta un sistema de ficheros raíz a partir de los contenidos de /dev/initrd. Durante la segunda fase, se cargan drivers y módulos adicionales a partir del contenido de sistema de ficheros inicial. Después de cargar los módulos adicionales, un nuevo sistema de ficheros (el sistema de ficheros raíz habitual) se monta desde un dispositivo diferente.

El proceso de arranque, cuando se utiliza “initrd”, es como sigue:

- 1) El gestor de arranque carga el núcleo y los contenidos de /dev/initrd en memoria.
- 2) Cuando comienza la ejecución del núcleo, éste se descomprime y copia el contenido de /dev/initrd en el dispositivo /dev/ram0, liberando posteriormente la memoria usada por /dev/initrd.
- 3) El núcleo monta el fichero /dev/ram0 como el sistema de ficheros raíz inicial, en modo lectura/escritura.
- 4) Si el sistema de ficheros raíz “normal” coincide con el sistema de ficheros raíz inicial (/dev/ram0), entonces el núcleo salta hasta el último paso de la secuencia de arranque habitual.
- 5) Si el ejecutable /linuxrc existe en el sistema de ficheros inicial, se ejecuta con uid<sup>4</sup> 0. Este fichero

---

<sup>3</sup> Basado en 'man initrd'

<sup>4</sup> UID=User Identifier o Identificador de Usuario que ejecuta un proceso. El UID 0 es el correspondiente al usuario *root* y cualquier proceso que se ejecute con este UID tiene todos los permisos *deroot*.

debe tener permisos de ejecución. `/linuxrc` puede ser cualquier ejecutable, incluyendo un script de shell.

- 6) Si `/linuxrc` no se ejecuta o bien ya ha terminado su ejecución, se monta el sistema de ficheros normal. NOTA: Si `/linuxrc` termina con algún sistema de ficheros montado en el sistema de ficheros raíz inicial, entonces el comportamiento del núcleo no está especificado.
- 7) Si el sistema de ficheros normal contiene un directorio llamado `/initrd`, entonces el dispositivo `/dev/ram0` se mueve de `/` a `/initrd`. En caso contrario, `/dev/ram0` es desmontado. NOTA: Cuando se mueve de `/` a `/initrd`, `/dev/ram0` no se desmonta y, por tanto, los procesos pueden continuar ejecutándose desde `/dev/ram0`. Si el directorio `/initrd` no existe en el sistema de ficheros normal y cualquier proceso continúa en ejecución desde `/dev/ram0` cuando `/linuxrc` termina, el comportamiento del núcleo no está especificado.
- 8) El proceso de arranque habitual (ejecución de `"init"`) tiene lugar en el sistema de ficheros normal.

### 3.El proceso de arranque en Meta-Distros<sup>5</sup>

En el caso de Meta-Distros, el proceso de arranque tiene lugar desde un CD-ROM. Para crear este CD-ROM de arranque se hace uso de la herramienta ISOLINUX, que es capaz de arrancar el sistema desde un CD-ROM y pasar posteriormente el control al proceso `"init"`.

Lo que vamos a analizar en este apartado es el proceso que tiene lugar desde que la BIOS pasa el control al CD-ROM hasta que pasa a ejecutarse el proceso `"init"`.

Todo este proceso se lleva a cabo a partir de un conjunto de herramientas, que se ha bautizado como *calzador*. El calzador incluye el conjunto de programas necesarios para hacer arrancar una distribución de GNU/Linux cualquiera desde un CD-ROM. La idea básica consiste en crear una distribución de GNU/Linux adaptada y ajustada a unas necesidades concretas y, a partir de la misma, crear una imagen de dicha distribución. Esta imagen, junto con las herramientas que forman parte del calzador, se copiarán a un CD-ROM, y será posible arrancar dicha distribución directamente desde el CD-ROM.

El calzador incluye, entre otras, la herramienta `isolinux` y todos sus ficheros de configuración. Es por ello por lo que el directorio en el que se encuentra el calzador en el CD-ROM se llama `/isolinux`.

#### 3.1 Configuración de ISOLINUX

A continuación se van a citar los ficheros de configuración del programa ISOLINUX. Se puede encontrar información más detallada acerca de los mismos en `"Creando un CD-ROM autoejecutable con ISOLINUX"` o en la documentación de ISOLINUX.

Los ficheros de configuración son los siguientes:

- `isolinux.bin`
- `isolinux.cat`
- `isolinux.cfg`

---

<sup>5</sup> Basado en 'Creando un CD-ROM autoejecutable con ISOLINUX'. Sergio González González. 2003.

- isolinux.msg
- logo.16
- f2.msg y f3.msg
- memtest
- spanish.kbd
- vmlinuz.usuario
- initrd.gz

Todos estos ficheros se encontrarán en el directorio /isolinux del CD final.

### **3.2 El archivo más importante: *initrd.gz*.**

Ya se ha descrito anteriormente la forma de arrancar un sistema cuando se utiliza “initrd”. La forma en que se debe crear este fichero se puede encontrar en “Creando un CD-ROM autoejecutable con ISOLINUX”.

Nosotros nos vamos a centrar en el proceso que tiene lugar cuando pasa a ejecutarse el contenido de este fichero.

En el caso de Meta-Distros, *initrd* debe ser capaz de preparar el sistema para arrancar la imagen de la distribución. Esto incluye cargar los módulos que sean necesarios, localizar la imagen de la distribución, etc. A lo largo de esta apartado, veremos cómo es posible conseguir todo esto.

#### **3.2.1 Estructura del contenido de *initrd* para Meta-Distros.**

##### ***El directorio '/'***

El directorio raíz contiene una estructura de directorios similar a la de cualquier sistema GNU/Linux, pero con algunas modificaciones. La estructura de directorios es la siguiente:

```

.
|-- META
|-- bin -> /META/bin
|-- boot -> /META/boot
|-- cdrom
|-- dev
|-- etc
|-- lib -> /META/lib
|-- linuxrc
|-- mnt
|-- modules
|-- opt -> /META/opt
|-- proc
|-- sbin -> /META/sbin
|-- static
|-- tmp -> /var/tmp

```

```
`-- usr -> /META/usr
```

Si lo comparamos con la estructura habitual de una distribución cualquiera vemos que hay directorios que faltan (como por ejemplo /home, /root, /floppy, /var...), hay alguno nuevo (/META) y hay muchos que son enlaces a otros dentro del directorio /META (salvo /tmp que apunta a /var/tmp; de esta manera se tienen unificados todos los directorios temporales, que necesitan permisos de escritura).

Es importante señalar la existencia del fichero /linuxrc, que, como sabemos, es el responsable del arranque del sistema una vez se ha montado initrd.

### ***El directorio /META***

El contenido del mismo es el siguiente:

```
.  
| -- bin -> /cdrom/META/bin  
| -- boot -> /cdrom/META/boot  
| -- etc -> /cdrom/META/etc  
| -- lib -> /cdrom/META/lib  
| -- opt -> /cdrom/META/opt  
| -- sbin -> /cdrom/META/sbin  
| -- usr -> /cdrom/META/usr  
`-- var -> /cdrom/META/var
```

El contenido de este directorio no es relevante puesto que, una vez localizada la imagen, se montará en este directorio, sobrescribiendo su contenido.

(NO ENTIENDO ENTONCES POR QUÉ EL CONTENIDO ES COMO ES)

### ***El directorio 'modules'***

En este directorio, se encuentran aquellos módulos imprescindibles para habilitar los diferentes métodos de arranque del sistema. Por tanto, deberá incluir módulos para dar soporte a aquellos dispositivos que pueden ser potencialmente utilizados para arrancar el sistema, como pueden ser dispositivos IDE, SCSI, USB o PCMCIA.

Aparte, debe existir el módulo “loop.o”. Éste permite montar en el loopback imágenes comprimidas de un sistema GNU/Linux. Este módulo es necesario para poder arrancar sistemas que hayan sido previamente comprimidos para optimizar el espacio y poder añadir más paquetes y programas a la metadistribución. De esta manera, se pueden crear metadistros que ocupen hasta 2 GB. en un solo CD.

### ***El directorio 'keymaps'***

Este directorio se nombra en “Creando un CD-ROM autoejecutable con ISOLINUX”, pero se comenta que no hay problema si no existe porque una vez esté cargado el sistema de la imagen, ya se incluyen estos mapas de teclado.

## **El directorio 'dev'**

Contiene los ficheros de dispositivos. No hay cambios significativos con respecto a un sistema normal, a excepción de que el número de dispositivos se ha reducido considerablemente, puesto que sólo se han incluido aquellos dispositivos que son imprescindibles para *initrd*.

## **El directorio 'etc'**

Dentro de este directorio existe un reducido grupo de ficheros de configuración:

```
.
| -- auto.mnt
| -- exports
| -- filesystems
| -- fstab
| -- group
| -- ld.so.conf -> /META/etc/ld.so.conf
| -- mtab
| -- passwd
| -- resolv.conf
|-- shadow
```

## **El directorio 'static': BusyBox**

*BusyBox* combina versiones pequeñas de muchas utilidades comunes UNIX en un único ejecutable pequeño. Provee un conjunto de reemplazos para la mayoría de utilidades básicas, como *grep*, *gzip*, *tar*, etc. Las utilidades de *BusyBox* suelen tener un menor número de opciones que las aplicaciones “reales”, pero mantienen las funcionalidades básicas.

*BusyBox* se ha diseñado con la idea de minimizar el tamaño y los recursos que ocupan estas aplicaciones al máximo.

Más información: En la documentación de *BusyBox*.

Para el caso de Meta-Distros, todas las utilidades compiladas en *BusyBox* se han ubicado en */static*. Estas utilidades son necesarias para los scripts de arranque puesto que, entre otras, ahí se encuentran utilidades como la *shell*.

Para acceder a las utilidades compiladas dentro de *BusyBox* directamente, es necesario crear enlaces “duros” desde cada una de las utilidades al ejecutable “*busybox*”, que se encargará de ejecutar estas aplicaciones adecuadamente.

El resultado de utilizar *BusyBox* es que se dispone de un conjunto mínimo de ejecutables básicos para cualquier sistema UNIX que permiten la ejecución de los diversos scripts de arranque en *initrd*.

## **El fichero '/linuxrc': el control del arranque en initrd.**

Este fichero es el responsable de la ejecución del arranque hasta que se le pasa el control al proceso “*init*”. Es, con mucho, el fichero más importante de todo el sistema *initrd*, y debe ser analizado en

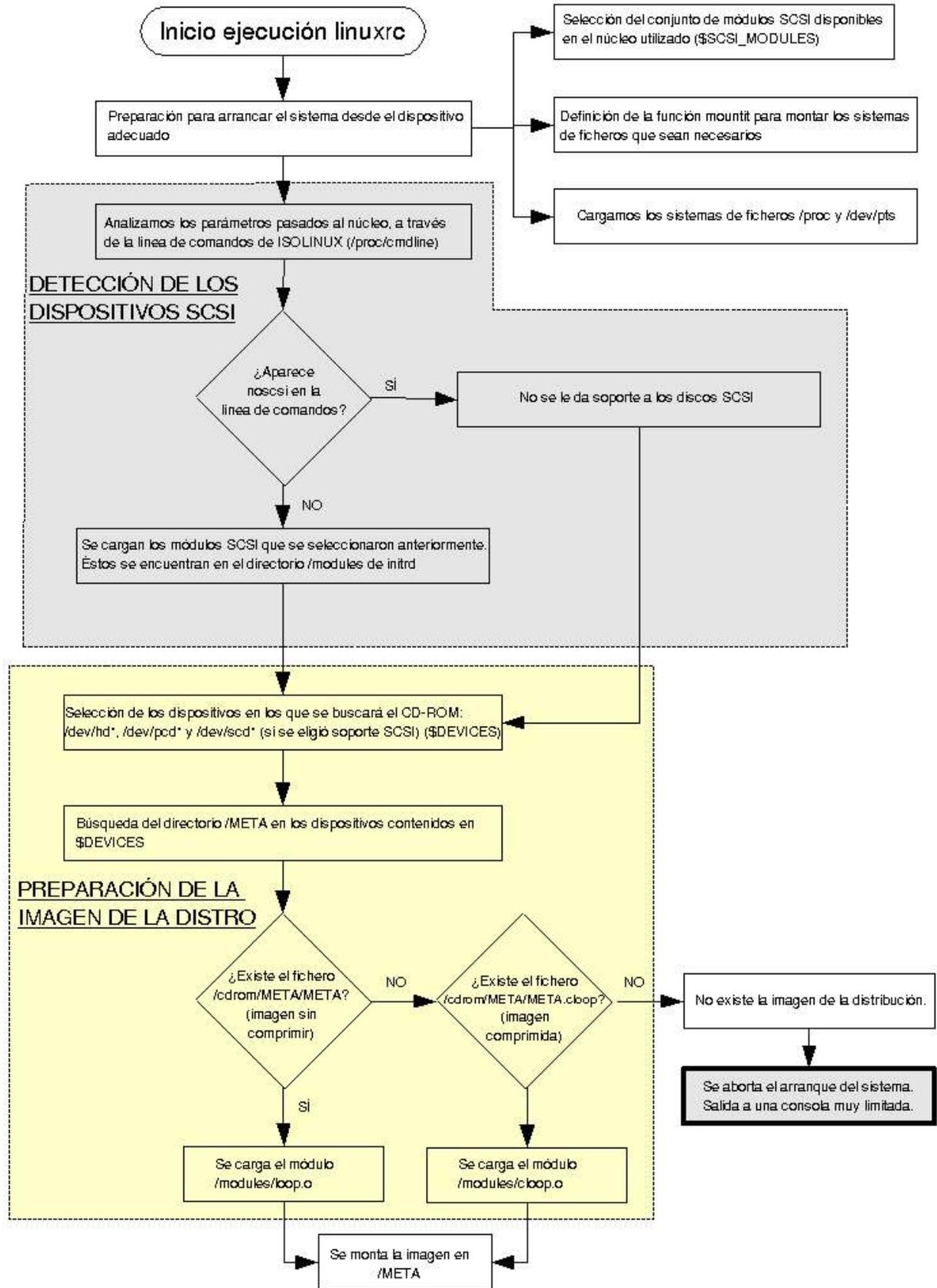
detalle.

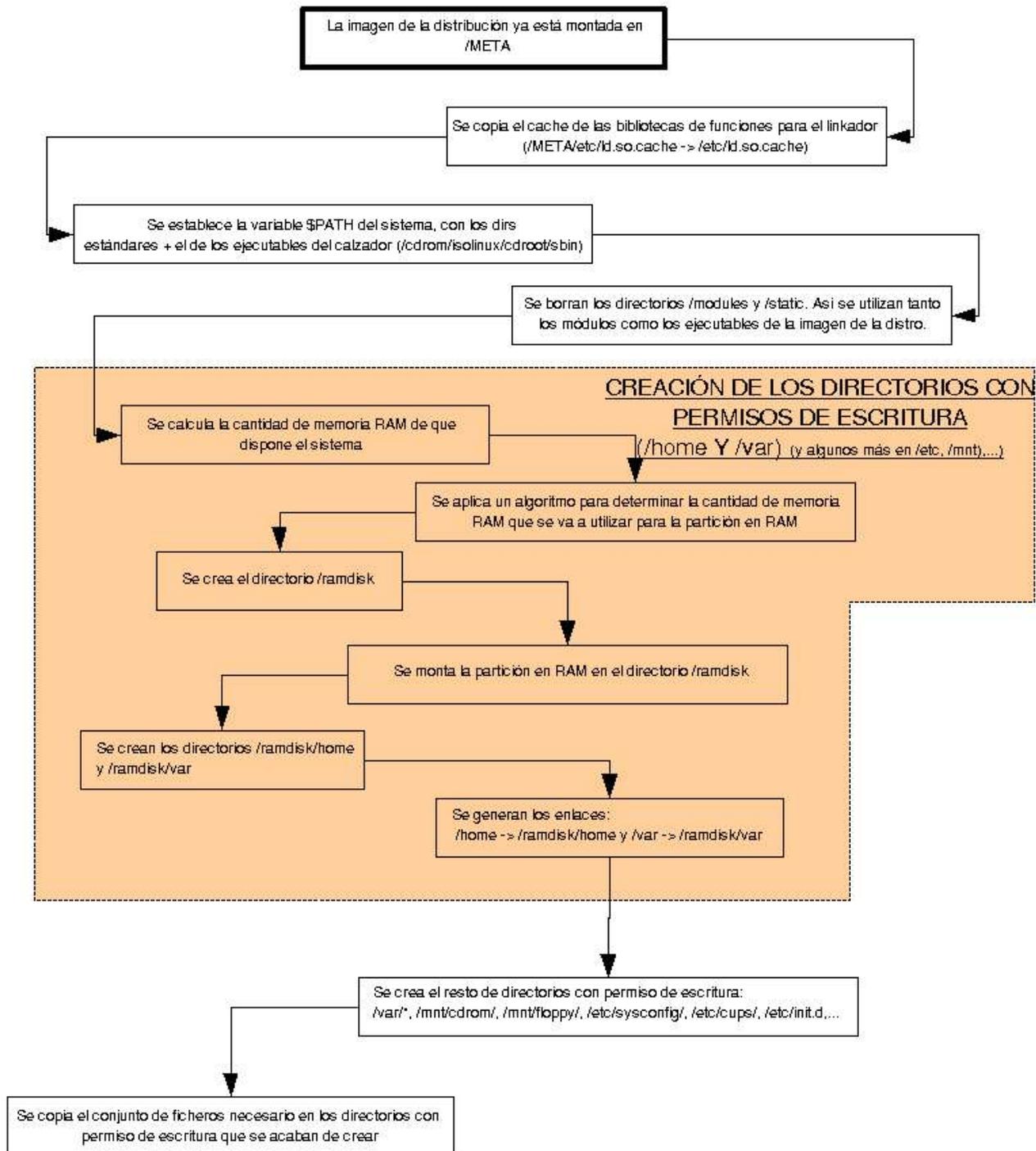
La ejecución de *linuxrc* tiene como objetivo principal preparar el sistema para arrancar la imagen de la metadistribución que hayamos creado. En realidad, éste es el objetivo del calzador, dentro del cual se encuentra toda la parte relacionada con *initrd* y *linuxrc*.

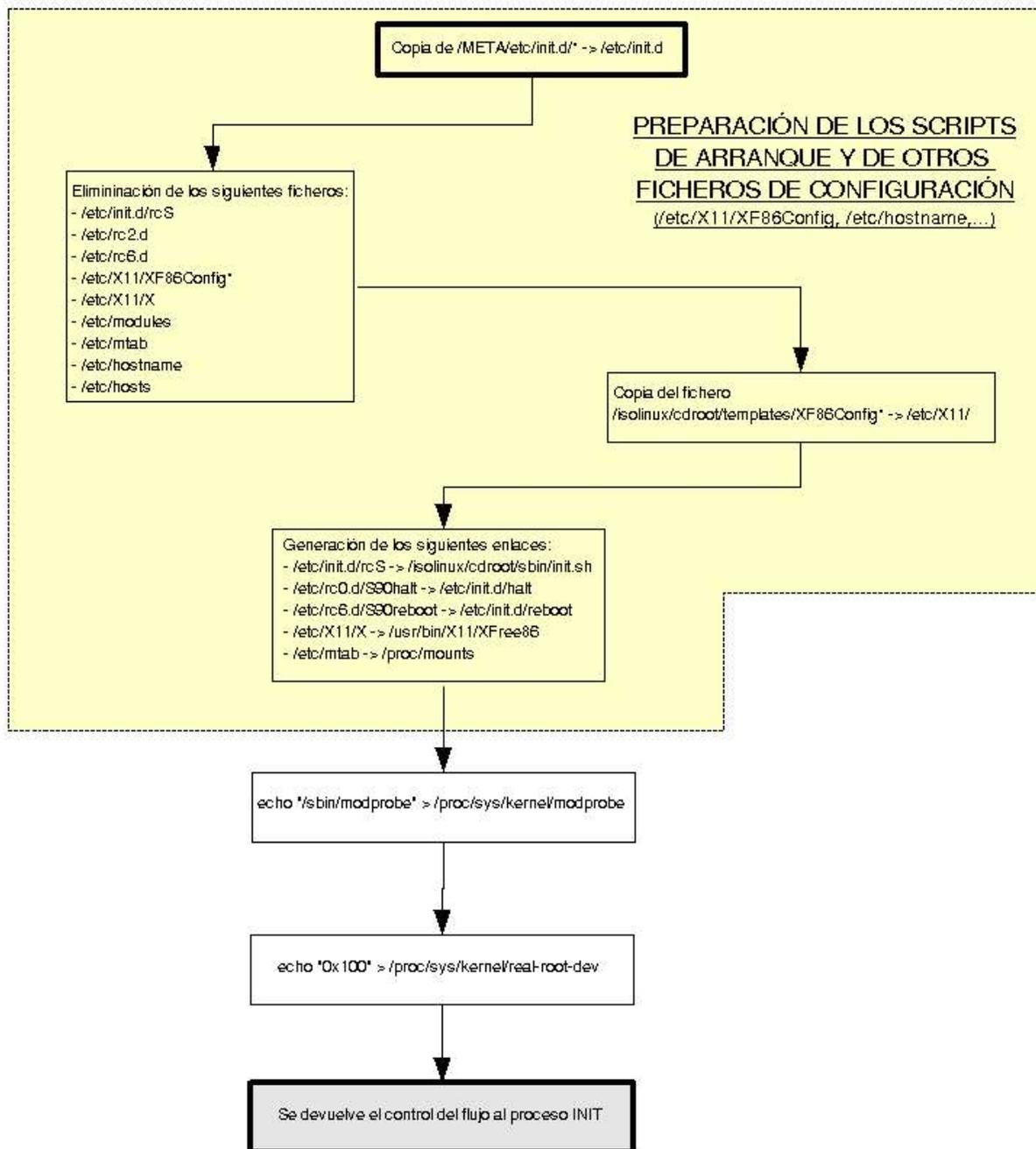
Entonces, el resultado de la ejecución de *linuxrc* debe ser un sistema preparado para trabajar con una metadistribución cualquiera que, en principio, no debe reunir ningún tipo de requisito ni condición especial para funcionar.

Una de las particularidades de todo esto es que se debe contemplar la posibilidad de ejecutar el sistema en modo *live*, lo que quiere decir que no se instala nada en el disco duro; toda la ejecución del sistema operativo tiene lugar desde una unidad de CD-ROM. La enorme cantidad de *hardware* disponible en el mercado, así como las dificultades que conlleva el disponer de un sistema de ficheros de solo lectura (como es el caso de un CD-ROM) convierten al calzador en un elemento delicado: será necesario llevar a cabo todo un conjunto de acciones concretas para hacer posible todo esto. Y esto es precisamente lo que vamos a analizar en este apartado.

Para comprender adecuadamente el funcionamiento de *linuxrc*, veamos el diagrama de flujo que aparece en las siguientes páginas, que muestra, a grandes rasgos, las diferentes acciones que lleva a cabo este script.







El proceso completo lo podemos estructurar, a grandes rasgos, en los siguientes pasos:

### 1) Preparación del sistema para arrancar desde la mayor cantidad posible de dispositivos de CD-ROM.

Esto incluye una gran variedad de dispositivos y buses a los que se conectan, incluyendo buses IDE, SCSI o incluso dispositivos que se conectan al puerto paralelo.

Para hacer posible lo anterior, es necesario disponer de los módulos del núcleo necesarios para dar soporte a toda esta variedad de *hardware*. En particular, serán necesarios los módulos SCSI. Éstos se encuentran en el directorio /modules del sistema de ficheros de *initrd*.

### 2) Detección de los dispositivos SCSI.

En primer lugar, se analiza si se pasó por la línea de comandos la palabra “noscsi”, que deshabilitaría la detección de este tipo de dispositivos.

Si no es el caso, se procede a cargar cada uno de los módulos de que se dispone. Si la carga de alguno tiene éxito quiere decir que existe el dispositivo al que da soporte.

### 3) Preparación de la *imagen* de la distribución.

El siguiente paso consiste en localizar la imagen de la distribución con la que se va a trabajar (que es la metadistribución propiamente dicha que hemos creado).

Para ello, se introduce en la variable \$DEVICES el conjunto de dispositivos que potencialmente pueden tener la imagen. Éstos incluyen a dispositivos IDE (/dev/hd?), SCSI (/dev/scd?) y CD-ROM por puerto paralelo (/dev/pcd?).

Entonces, se procede a intentar montar todos estos dispositivos. Si hay éxito, se comprueba la existencia del directorio META dentro del mismo (es decir, el directorio /cdrom/META).

Si existe este directorio, la imagen ha sido localizada. Lo que falta es determinar si la imagen está comprimida (con *cloop*) o no. Para ello, se hace una búsqueda del fichero “META” o “META.cloop” dentro del directorio /cdrom/META. Si no se encontrara ninguno de los dos ficheros, se abortaría el proceso de arranque y se saldría a una consola muy limitada.

En caso contrario, se montaría la imagen en /META.

### 4) Algunos detalles más.

Se realizan algunas tareas como copiar el caché de las bibliotecas de funciones para el linkador (ESTO NO SÉ POR QUÉ SE HACE).

Además, se establece la nueva variable del PATH con los directorios habituales de cualquier distribución (/bin, /sbin,...) más el directorio de los binarios del calzador (/cdrom/isolinux/cdroot/sbin). Inicialmente, esta variable se inicializó únicamente con el valor “/static”, que era la ruta donde se encontraban los binarios necesarios en la primera fase.

Por último, se borran los directorios “/modules” y “/static”, que ya no son necesarios. De esta manera, a partir de ahora se utilizarán los módulos y ejecutables de la imagen montada.

## 5) Creación de los directorios con permiso de escritura.

En cualquier sistema GNU/Linux es necesario disponer de algunas partes del sistema de ficheros con permisos de escritura. Esto es imprescindible para que determinados programas puedan escribir cierta información. Pero también es necesario para que un usuario pueda acceder en escritura al disco.

En el caso habitual de una distribución instalada en el disco duro, dada la naturaleza innata de este dispositivo, es posible acceder en escritura al sistema de ficheros.

Sin embargo, en el caso que nos ocupa, debido a que la imagen de la distribución se encuentra en un dispositivo de sólo lectura (como es el caso de un CD-ROM) y a que uno de los objetivos de Meta-Distros es no tener que modificar absolutamente ni un bit del disco duro del usuario, hay que buscar algún método para habilitar la escritura en el sistema de ficheros.

La solución pasa por utilizar parte de la memoria RAM como una unidad donde montar una parte del sistema de ficheros.

El primer paso consiste en determinar la cantidad de memoria RAM con la que cuenta el sistema. Conocido este dato, es necesario decidir qué parte de la misma se va a destinar para la partición.

Esta partición se montará en el directorio `/ramdisk`, dentro del cual se crearán los directorios “home” y “var”. El siguiente paso es crear sendos enlaces de `/home` y de `/var` a los correspondientes en “`/ramdisk`”.

Es importante señalar que serán necesarios más directorios con permisos de escritura fuera de estos dos: `/etc/sysconfig`, `/etc/cups/`,... Si recordamos, *initrd* se montó precisamente en RAM y, por tanto, es posible escribir en el mismo, así que lo único que hemos de hacer es crear dichos directorio en el sistema de ficheros.

Una vez creados todos los directorios con permiso de escritura, lo único que falta es copiar todos los ficheros necesarios a los mismos.

## 6) Preparación de los scripts de arranque y de otros ficheros de configuración.

El siguiente paso durante la ejecución de *linuxrc* es preparar el arranque para cuando el proceso INIT tome el control. Esto involucra fundamentalmente a los directorios `/etc/init.d` y `/etc/rc*.d`.

Lo que se hace es, en primer lugar, eliminar todos los ficheros que afectan al arranque para, posteriormente, “reconstruir” el sistema de arranque como interese.

En este sentido, lo que se hace es regenerar el fichero `/etc/init.d/rcS`, que se hace apuntar al script “`init.sh`” que se encuentra en el calzador. De esta manera, queda modificado la secuencia habitual de arranque, para pasar a una en la que se tiene en cuenta todas las condiciones especiales del sistema del que estamos hablando (este proceso se comentará posteriormente).

Aparte, se crean otros enlaces relacionados con el arranque, como se puede ver en el diagrama de flujo.

Asimismo, se copia del calzador una plantilla del fichero de configuración de X-Window al directorio `/etc/X11`.

## 7) Detalles finales.

Por último, se habilita la capacidad del núcleo para cargar módulos, se cambia la ubicación del

sistema de ficheros raíz (ESTO NO LO ENTIENDO MUY BIEN) y se devuelve el control a INIT, que pasa a ejecutar el fichero `/etc/init.d/rcS`.

### **3.3 Continuación del arranque: el proceso INIT**

Una vez que termina de ejecutarse el script `linuxrc`, el control del arranque pasa a INIT.

Como ya se ha comentado, este programa analiza el fichero `/etc/inittab` en busca de la línea `sysinit`, que indica el script que debe ejecutar.

En el caso que nos ocupa, se procederá a ejecutar `/etc/init.d/rcS`. Habitualmente este script lanza la ejecución de todos los scripts ubicados en `/etc/rcS.d/`. Sin embargo, en el caso de metadistros, el fichero `/etc/init.d/rcS` es un enlace al script “`init.sh`” que se encuentra en el directorio de los binarios del calzador.

Por tanto, cuando el proceso INIT toma el control del arranque, lo que sucede realmente es que se produce la ejecución de `/cdrom/isolinux/cdroot/sbin/init.sh`.

El proceso que tiene lugar se muestra en el diagrama de flujo que aparece en las páginas siguientes.

#### **3.3.1 Análisis del diagrama de flujo.**

El proceso que tiene lugar se puede esquematizar en los siguientes pasos:

1) El primer paso que tiene lugar es la carga de los ficheros de configuración de Meta-Distros. Éstos se encuentran en el directorio `/cdrom/isolinux/conf`. Cada uno de ellos contiene diferentes variables que controlan diversos aspectos del funcionamiento de la metadistribución, una vez se está ejecutando.

Estos ficheros de configuración son los siguientes:

- **q.conf:** Aquí se define el conjunto de variables que aparece en el diagrama de flujo. El significado de las mismas es el siguiente:
  - **QSPLASH:** Indica si se muestra o no la pantalla de “Splash” durante la carga del sistema, pantalla que “oculta” los mensajes del núcleo y muestra algún tipo de indicador de carga, con la idea de convertir el arranque en algo más divertido.
  - **QLANGUAGE:** Controla si se le pregunta o no al usuario por el lenguaje que debe utilizar para la distribución.
  - **QPASS:** Indica si se pregunta al usuario por la contraseña de *root*.
  - **QUSER:** Controla si se pregunta o no por el nombre del usuario del sistema y su contraseña (usuario que existirá además de *root*).
  - **QHOST:** Indica si se pregunta por el nombre del equipo o no.
  - **QINSTALL:** Controla si se pregunta al usuario si quiere instalar la distribución al disco duro.
  - **QX:** (NO ESTOY SEGURO; ANALIZAR)
  - **QEXPERT:** (NO ESTOY SEGURO)
  - **QNET:** Indica si se pregunta o no por la configuración manual de los parámetros de red.

- **var.conf**: En este caso, las variables que se definen son las siguientes:
    - DISTRO: Controla el nombre de la distribución.
    - USERNAME: Nombre del usuario alternativo a *root*.
    - UPASSWORD: Contraseña del usuario anterior.
    - RPASSWORD: Contraseña de *root*.
    - HOSTNAME: Nombre del equipo.
    - LANGUAGE: Lenguaje que se utilizará.
    - INSTALL: Indica si se debe instalar la distribución o se debe hacer un arranque *live*.
    - ROOT: Controla si se crea o no un usuario alternativo a *root*.
    - EXPERT: Si se activa, se pueden seleccionar opciones más complejas.
    - XCONF: Esta opción está pensada para monitores y tarjetas gráficas antiguas.
    - DHCP: Activada, indica una configuración de la red mediante DHCP.
    - STARTUSER: Nombre del usuario con el que arrancará la distribución.
    - STARTX: Indica si se arranca o no el servidor X cuando arranque la distribución.
    - LTSP: Controla si se activa o no el servicio LTSP<sup>6</sup>
  
  - **lang.conf**: Este último fichero de configuración contiene información acerca de las “locales” específicas para cada idioma. Éstas son las variables que contienen la información relativa al lenguaje que se va a utilizar, el tipo de moneda, y demás información regional propia de cada país.
- 2) El siguiente paso consiste en analizar lo que se ha introducido por la *línea de comandos*. Es posible pasar comandos al núcleo antes de arrancar el sistema<sup>7</sup>. También es posible pasar a través de la línea de comandos las variables anteriores con un valor diferente del predeterminado. Este valor introducido prevalece frente al que se estableció por defecto. Así, se modifica el comportamiento de la metadistribución en relación con el habitual.
  - 3) Ahora se procede a configurar adecuadamente el teclado. Para ello, se analiza la variable KEYTABLE del fichero “lang.conf” y, en función del valor de la misma, se carga la tabla de teclas adecuada.
  - 4) El siguiente paso es establecer el PATH. Se introducen las rutas habituales en cualquier sistema GNU/Linux, sin incluir las relacionadas con X-WINDOW, y añadiendo el directorio de los binarios del calzador.
  - 5) Llegados a este punto, se produce el inicio de la *detección de hardware* (a partir de la ejecución del script “hw-detect.sh”). Este proceso tendrá lugar en segundo plano, mientras continúa la ejecución del script principal. Analizaremos en detalle el proceso de detección de hardware en los apartados siguientes.
  - 6) Lo siguiente que tiene lugar es el análisis de las diferentes variables que se encontraron en los

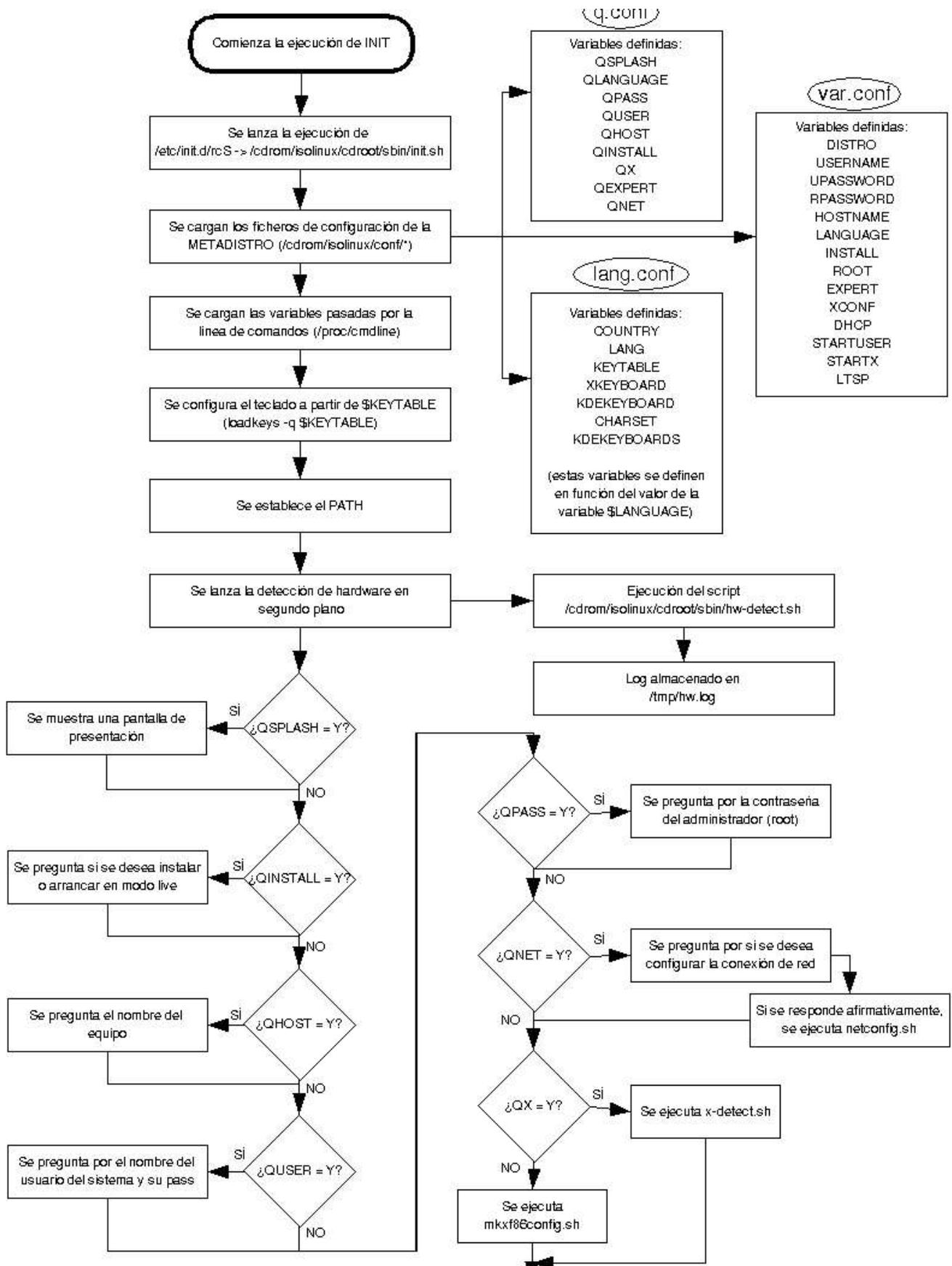
<sup>6</sup> LTSP = Linux Terminal Server Project

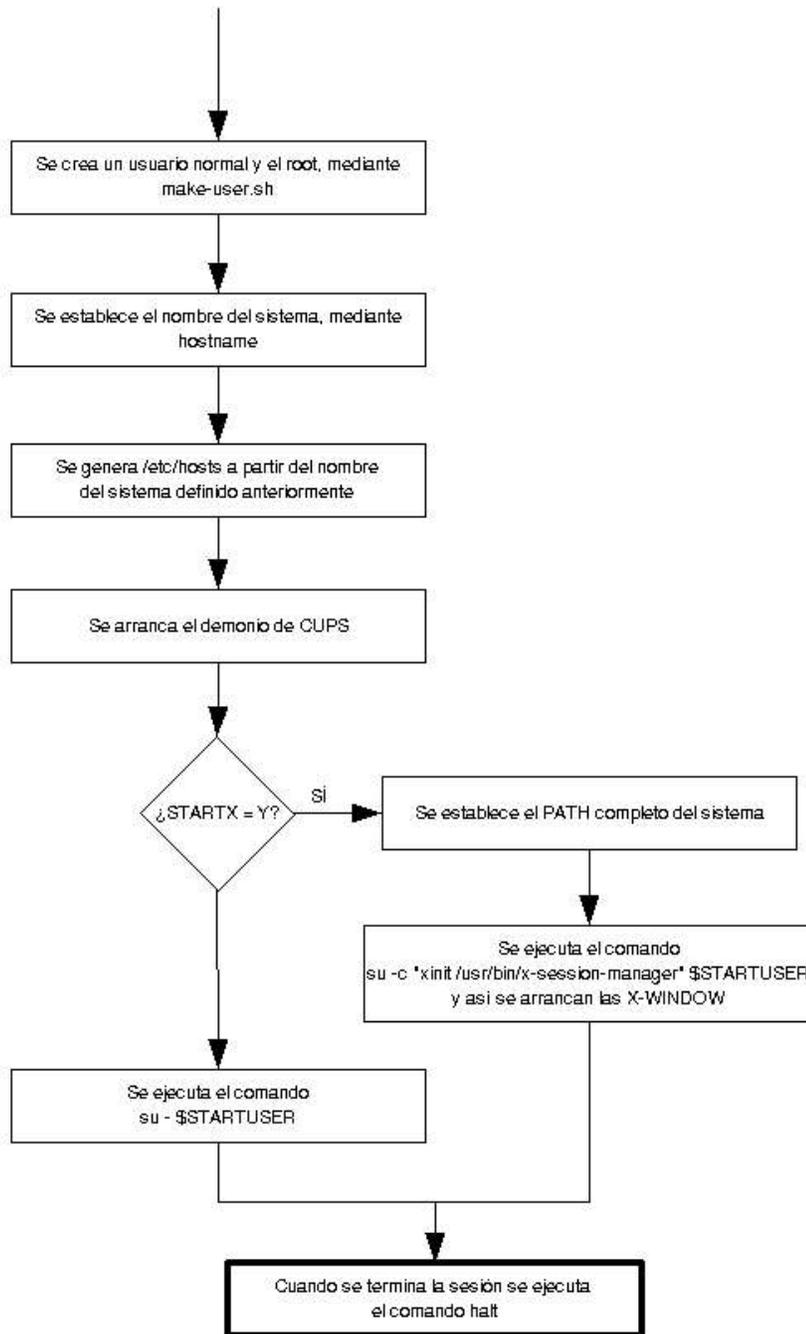
<sup>7</sup> Ver “BootPrompt HOWTO”.

ficheros de configuración para realizar las acciones pertinentes. Así pues, por poner un ejemplo, si la variable QSPLASH tiene un valor verdadero (o “Y”), se mostrará una pantalla de presentación durante el arranque.

Es importante hacer hincapié en dos variables particulares:

- QNET: Si esta variable contiene un valor verdadero, se procederá a configurar la red manualmente a partir de la ejecución del script “netconfig.sh”.
  - QX: En este caso, si la variable es verdadera se ejecutará “x-detect.sh”. En caso contrario, se ejecutará “mkxf86config.sh”.
- 7) Una vez analizadas todas las variables de configuración, se procede a crear un usuario normal y el root, a partir de la ejecución de “make-user.sh”.
  - 8) Se continúa el arranque estableciendo el nombre del sistema, generando el fichero /etc/hosts a partir de la información anterior, y se arranca el demonio CUPS.
  - 9) Por último, si la variable STARTX es verdadera, se arrancará X-WINDOW. En este caso, se ampliará el contenido de la variable PATH, introduciendo las rutas de los binarios de X, y se ejecutará el comando xinit con los parámetros adecuados, para lanzar la sesión de X. Si STARTX es falsa, el arranque termina en una consola de usuario.
  - 10) En cualquiera de los dos casos anteriores, una vez que la sesión termina, se produce la ejecución del comando “halt”, que cierra el sistema.





### 3.3.2 El proceso de detección de Hardware: hw-detect.sh.

Durante el proceso de arranque del sistema (mientras se ejecuta “init”), tiene lugar la ejecución en segundo plano del script “hw-detect.sh”.

Este script es el verdadero responsable de la detección de hardware en la metadistribución.

Vamos a analizar en este apartado los diferentes pasos que tienen lugar durante la ejecución del mismo:

- 1) En primer lugar, se establece el reloj del sistema a partir del reloj hardware, mediante el comando “hwclock”. Se le pasan las opciones “-s” (para establecer la hora del sistema a partir de la del reloj hardware) y “-utc” (para indicar que la hora se almacena en formato “Coordinated Universal Time” y no en formato local).<sup>8</sup>  
(¿NO ES MÁS LÓGICO SUPONER FORMATO LOCAL EN LUGAR DEL OTRO?)
- 2) Se monta el sistema de ficheros raíz como lectura y escritura.
- 3) Se da soporte al sistema para APM, cargando el módulo “apm.o”. Se pasa el parámetro “power\_off=1” para permitir que APM apague el sistema.
- 4) El siguiente paso es dar soporte para dispositivos PCMCIA. Para ello, se carga el módulo principal: “pcmcia\_core.o”. Y después se intenta cargar los módulos siguientes:
  1. yenta\_socket.o: soporte para CardBus (la nueva versión de 32 bits de las tradicionales tarjetas PCMCIA de 16 bits).
  2. i82365.o: soporte para “bridges” para bus ISA, de equipos antiguos.
  3. tcic.o: soporte para unos “bridges” especiales de equipos antiguos.

Así se consigue habilitar el soporte tanto para tarjetas CardBus como para hardware antiguo especial.

Si la carga de ninguno de estos módulos tiene éxito, se descarga el módulo “pcmcia\_core” y se establece PCMCIA=0.

En caso contrario, se carga el módulo “ds” (que también forma parte del soporte básico de PCMCIA junto con el módulo “pcmcia\_core”) y se ejecuta el comando “cardmgr”. Así queda habilitado el soporte para dispositivos PCMCIA.

- 5) Ahora se pasa al soporte de dispositivos USB. El procedimiento es similar al anterior. En primer lugar, se carga el módulo “usbcore.o”, que proporciona el soporte básico para dispositivos USB. Después, es necesario dar soporte al hardware USB específico que existe en el sistema. Para ello, hay que cargar uno de los módulos siguientes: “usb-uhci.o” o “usb-ohci.o” (HABRÍA QUE AÑADIR DOS MÓDULOS MÁS: uhci.o (no sé qué diferencias hay con el anterior) y ehci-hcd.o (para USB 2.0)).

En este punto, y suponiendo que los módulos se hayan cargado sin problemas, se monta el sistema de ficheros usbdevfs, para permitir, mediante /proc/, el acceso a la información de los dispositivos USB conectados.

---

<sup>8</sup> Más info: man hwclock

<sup>9</sup> Más info: paquete pcmcia-es

Sin embargo, aún no existe soporte para ningún dispositivo USB concreto. Sólo está cargada la base común para todos los dispositivos USB. La carga de los módulos concretos según el hardware que haya conectado tendrá lugar más adelante.

Si la carga de los módulos ha tenido problemas, se descarga el módulo “usbcore.o”.

- 6) Ahora se procede a proporcionar soporte para dispositivos FIREWIRE. Lo único que se hace es cargar el módulo “ohci1394.o”, que da soporte para un conjunto de chipsets muy utilizados. (ALGO EXTRAÑO: en este caso no se carga el módulo del “core” de FIREWIRE como en los casos anteriores ¿¿¿¿???) .
- 7) El siguiente paso es dar soporte para HOTPLUG, en función de lo que se haya detectado hasta el momento. Hay una variable, HOTPLUG, que inicialmente está vacía. Esta variable pasará a contener “yes” siempre y cuando se detecte algún dispositivo USB, FIREWIRE o PCMCIA (ALGO EXTRAÑO: en el caso de PCMCIA, sólo si se carga satisfactoriamente el módulo “yenta\_socket”, se activa esta variable).

En el caso de que esta variable contenga “yes”, se activa el manejador “HOTPLUG” (/sbin/hotplug)<sup>10</sup>.

- 8) Detección del resto de dispositivos hardware.

Para ello, se utiliza la herramienta “hwsetup”. Este programa, escrito por Klaus Knopper, es básicamente una modificación de Kudzu, que utiliza las mismas librerías de detección de hardware, pero que funciona de manera levemente diferente. En este caso, la detección y configuración de hardware se produce de manera no interactiva. Además, se cargan los módulos necesarios y se generan los ficheros en /dev<sup>11</sup>.

El funcionamiento de hwsetup es muy similar al de Kudzu: hace una búsqueda del hardware instalado en el sistema y compara los resultados con una base de datos de dispositivos que se almacena en /cdrom/isolinux/cdroot/hwdata (en el calzador). A partir de la misma se cargan los módulos necesarios para dar soporte al hardware detectado.

(AMPLIAR LA DOCUMENTACIÓN)

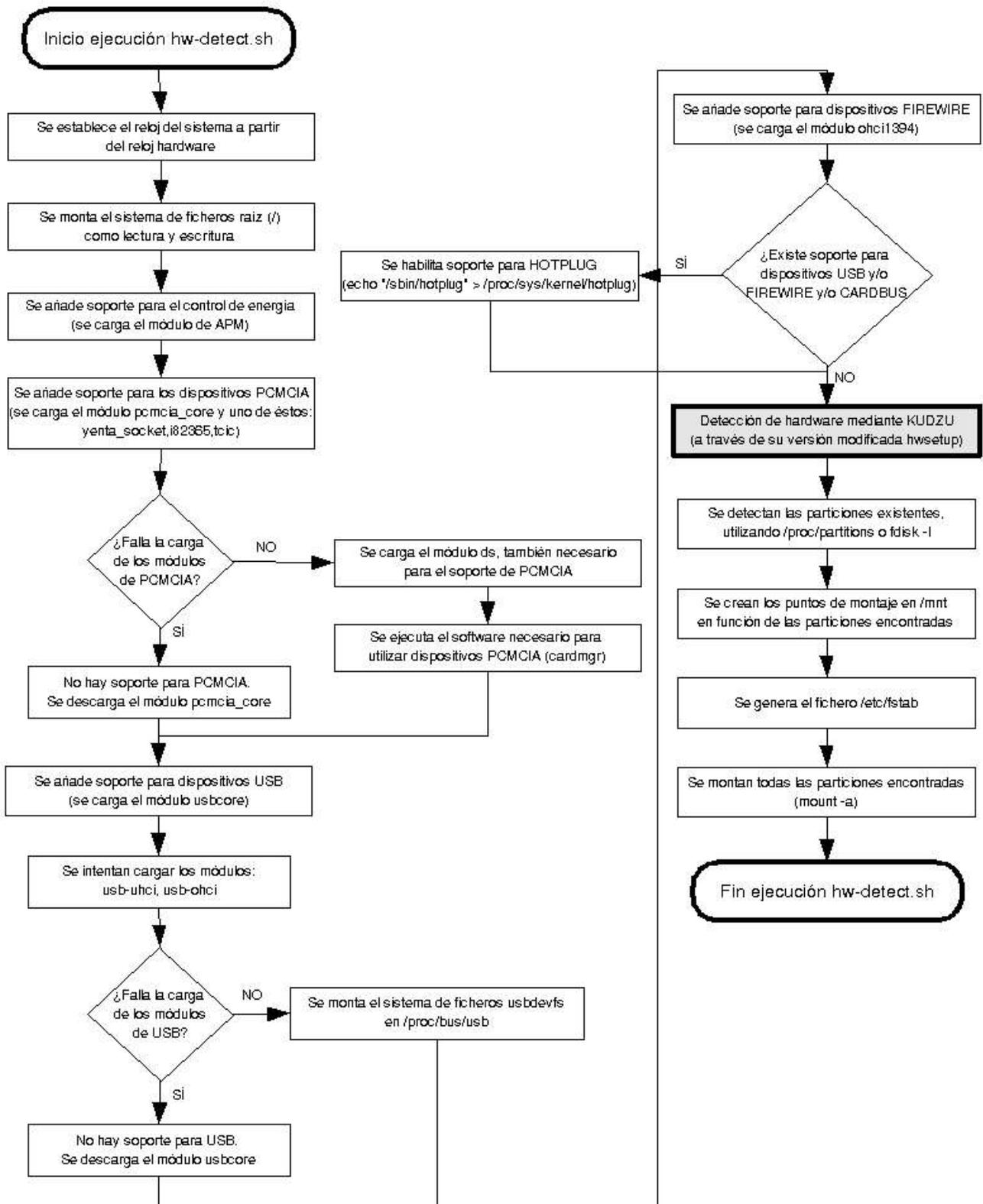
- 9) El último paso consiste en detectar y montar todas las particiones existentes. Para ello, se hace una búsqueda en /proc/partitions. A partir del resultado y con la ayuda de fdisk se dispone de toda la información acerca de las particiones y del tipo de sistema de ficheros que hay en cada una de ellas. Entonces, se crean los puntos de montaje en /mnt, se genera el fichero /etc/fstab y se montan todas las particiones encontradas.

- 10) Se termina la ejecución de hw-detect.sh.

---

10 Más info: Paquete “hotplug”.

11 Más info: <http://www.knopper.net/download/knoppix/>, donde se encuentra el código fuente de “hwsetup”.



## **4.El sistema de detección de Hardware en MetaDistros.**

En esta apartado, se pretende analizar, de manera global, en qué puntos, dentro del arranque de una metadistribución, tiene lugar algún tipo de detección automática de *hardware*. Esto ya se ha visto en apartados anteriores. Se trata simplemente de resumir brevemente en qué puntos se trata de hacer algún tipo de detección, para tener una visión amplia de este proceso, y poder actuar sobre él para mejorarlo. Asimismo, se pretende dar también una idea de cómo añadir soporte para nuevos dispositivos.

El proceso de detección de *hardware* en Meta-Distros tiene lugar en dos fases claramente diferenciadas: una primera, en la que tiene lugar la carga de aquellos módulos necesarios para cargar la imagen de la metadistribución; y una segunda, en la que se produce la detección de todo el resto de dispositivos *hardware*.

### **4.1Primera fase: *initrd*.**

Esta primera fase tiene lugar durante la ejecución de *initrd* y, como se ha comentado, su único objetivo es habilitar el soporte para todos aquellos dispositivos que pueden potencialmente contener la imagen del sistema. En este caso, se utilizan los módulos que existen en el directorio `/modules` de *initrd*. Básicamente se trata de módulos para dispositivos SCSI (además de *cloop* y *loop*).

La detección de *hardware* en esta primera fase es un poco rudimentaria, en el sentido de que se utiliza el método de “prueba y error”. Primero se prueba a cargar todos los módulos SCSI probando a ver si la carga de alguno de ellos tiene éxito, lo que indicaría que se ha detectado una controladora de ese tipo. Después, se prueba a montar todos los dispositivos de CD-ROM IDE, SCSI y por puerto paralelo en busca de alguna prueba que no falle. Si esto ocurre, se procede a buscar el directorio `/META` en dicho dispositivo. Y así es como se localiza la imagen de la distribución.

### **4.2Segunda fase: *proceso init*.**

La segunda fase tiene lugar cuando se ejecuta el proceso “*init*”. En este caso, la detección de *hardware* corresponde al script `hw-detect.sh`.

En este caso, y resumiendo mucho, durante la primera parte, se trata de dar soporte a dispositivos PCMCIA, USB y FIREWIRE. Hecho esto, se utiliza un programa derivado de Kudzu, que se encarga de buscar y cargar los módulos adecuados para el resto de dispositivos existentes en el sistema.

### **4.3Cómo añadir soporte para nuevos dispositivos.**

(ANALIZAR ESTE PUNTO EN PROFUNDIDAD)

## 5. Funcionamiento de KUDZU.

(MADURAR ESTE APARTADO UN POQUITO)

### 5.1 Descripción

Kudzu es un paquete de detección de hardware, desarrollado por Red Hat e incluido por primera vez en la versión 6.1, que engloba a un conjunto reducido de ejecutables: kudzu, updfstab y module\_upgrade. Dentro de dicho paquete, existen, asimismo, manuales de ayuda para cada uno de estos ejecutables.

Por otra parte, existen otros dos paquetes de software relacionados, llamados kudzu-dev y kudzu-vesa:

- kudzu-dev: Conjunto de cabeceras (.h) y librerías para enlazar programas que utilicen este sistema de detección de HW.
- Kudzu-vesa: Conjunto de cuatro ejecutables: ddcprobe, ddcxinfo, modetest y svgamodes. Estos programas están orientados a la detección de dispositivos gráficos instalados en el sistema y al análisis de los modos gráficos soportados por los mismos.

Por último, existe un último paquete relacionado (hwdata), que provee a kudzu de la base de datos de dispositivos sobre la que trabajar. Incluye la identificación hardware y los datos de configuración de distintos dispositivos, a partir de ficheros como pci.ids o la base de datos de tarjetas para Xfree86.

### 5.2 Funcionamiento

Cuando kudzu se ejecuta, lo primero que se hace es detectar el hardware presente en el sistema, y lo compara con el contenido del fichero /etc/sysconfig/hwconf, si es que existe. Entonces determina si se ha añadido y/o quitado algún dispositivo hardware del sistema. Si esto ocurre, le brinda al usuario la posibilidad de configurar el hardware que se haya añadido y borra la configuración de aquellos dispositivos que hayan sido eliminados.

Si no existe ninguna base de datos del hardware del sistema, kudzu intenta determinar aquellos dispositivos que ya hayan sido configurados. Para ello, analiza el contenido de /etc/modutils/kudzu, /etc/sysconfig/network-scripts/ y /etc/X11/XF86Config.

Cuando Kudzu se encuentra “Configuring device X...” en realidad lo que está haciendo es lo siguiente:

- Red:
  - Añade un alias “ethX”, si es necesario.
  - Migra la configuración del dispositivo antiguo o crea uno nuevo.
- SCSI:
  - Añade un alias “scsihostadapter”.
- Tarjeta de vídeo:
  - Ejecuta “Xconfigurator”.

- Tarjeta de sonido:
  - Ejecuta “sndconfig”.
- Ratón:
  - Crea el enlace “/dev/mouse”.
  - Ejecuta “mouseconfig”.
- CD-ROM:
  - Genera el enlace “/dev/cdrom”.
- Scanner:
  - Genera el enlace “/dev/scanner”.
- Teclado:
  - Ejecuta “kbdconfig”.
  - Para la consola serie, se asegura de que “/etc/inittab” y “/etc/securetty” están preparados para operaciones de la consola serie.

Cuando se trata de eliminar la configuración de algún dispositivo, lo que se hace básicamente es eliminar los enlaces y alias que se crearon anteriormente.

### **5.3 Opciones**

--usage

Indica de manera breve cómo se utiliza el programa.

--help

Muestra la ayuda.

-q, --quiet

Realiza sólo aquella configuración que no requiere intervención por parte del usuario.

-s, --safe

Realiza sólo pruebas “seguras” del hardware. Esto deshabilita las pruebas del puerto serie, la prueba del monitor DDC y la prueba del ratón PS/2.

-t, --timeout [segundos]

Establece el temporizador para el diálogo inicial. Si se sobrepasa este tiempo sin tocar ninguna tecla,

kudzu sale y no se actualiza el contenido de /etc/sysconfig/hwconf.

-k, --kernel [versión]

Para determinar si un módulo existe, busca en esta versión de kernel especificada. Si no se especifica ninguna, se busca en la versión en uso del kernel.

-b, --bus [bus]

Hace pruebas de hardware sólo en el bus especificado, que puede ser SCSI, PCI, MISC,...

-c, --class [class]

Hace pruebas de hardware sólo para la clase especificada, que puede ser NETWORK, AUDIO, VIDEO,...

-f, --file [archivo]

Lee la información de hardware de un fichero y no realiza ninguna prueba de hardware.

-p, --probe

Muestra el resultado de la prueba de hardware por pantalla, pero no realiza la configuración de ningún dispositivo.

## **5.4 Ficheros relacionados**

### **5.4.1 /etc/sysconfig/hwconf**

Listado del conjunto de hardware detectado por Kudzu

### **5.4.2 /etc/sysconfig/kudzu**

Configuración para la prueba de hardware durante el arranque. Este fichero no existe y, en el momento actual, desconocemos el formato.

### **5.4.3 /etc/modutils/kudzu**

Fichero de configuración de los módulos.

#### **5.4.4/etc/sysconfig/network-scripts/ifcfg-\***

Ficheros de configuración de interfaces de red.

---

(PROBAR A HACER UN NÚCLEO MUCHO MÁS MODULAR PARA COMPROBAR LAS ACCIONES QUE LLEVA A CABO KUDZU)

## 6. Bibliografía

# 7. GNU Free Documentation License

GNU Free Documentation License

Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

## 0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

## 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is

available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

## 2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

## 3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain

thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

#### 4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the

authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

## 5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

## 6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

## 7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

## 8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers,

provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

## 9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

## 10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (c) YEAR YOUR NAME.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under

your choice of free software license, such as the GNU General Public License, to permit their use in free software.