

# Administración de Servidores

Arranque del Sistema

Cargadores de Arranque Alternativos

Creado por

Iñaki Fernández de Viana y González

Huelva, octubre 2020

# Sobre Nosotros

## Iñaki Fernández de Viana y González



Despacho 128. Escuela Técnica  
Superior de Ingeniería



Dpto. De Tecnologías de la Información



[i.fviana@dti.uhu.es](mailto:i.fviana@dti.uhu.es)



+34 959217378



# Objetivos

- **Peso:** 2
- **Descripción:** Candidates should be aware of other bootloaders and their major features.

# Objetivos (II)

- **Áreas clave de conocimiento:** SYSLINUX, ISOLINUX, PXELINUX. Understanding of PXE for both BIOS and UEFI. Awareness of systemd-boot and U-Boot.

# Objetivos (III)

- **Términos y utilidades:** syslinux. Extlinux. Isolinux.bin. Isolinux.cfg. Isohdpx.bin. Efiboot.img. Pxelinux.0. pxelinux.cfg/. uefi/shim.efi. uefi/grubx64.efi

# Índice

1. Introducción
2. LILO
3. Syslinux
4. Otros

# Introducción

---

1

---



# Introducción

- ★ Existen gran cantidad de gestores de arranque
- ★ Cada uno de ellos da solución a distintas casuísticas y arquitecturas.
  - ¿Cómo crear un CDROM autoarrancable?
  - ¿Cómo arrancar remotamente un núcleo?



# LILO

---

2

---

# Introducción

- ★ **LILLO** (Linux LOader) fue lanzado en 1992 para reemplazar a loadlin.
- ★ Fue el cargador de arranque de la mayoría de las distribuciones hasta 2008 cuando paró su desarrollo hasta 2010.
- ★ Principales diferencias con GRUB
  - El interfaz de interacción de GRUB es más completo
  - GRUB es un proyecto vivo
  - LILO requiere reescribir el MBR cada vez que se modifica su fichero de configuración

# Configuración

- ★ El funcionamiento de **LILO** se rige por el fichero **/etc/lilo.conf**.
- ★ Disponemos de una sección de **opciones globales**:

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
compact
default=Linux
```

## Configuración (II)

★ ... y opción por núcleo

```
image=/boot/vmlinuz-2.6.23-13  
label=Linux  
read-only  
root=/dev/hdb3  
password=Linux
```

```
image=/boot/vmlinuz-2.4.18-14  
label=ker2418  
read-only  
root=/dev/hdb3
```

```
other = /dev/hda  
label = WindowsXP
```

# Opciones globales

Opción	Descripción
<code>boot</code>	The target drive or partition
<code>map</code>	Sector address map file to be created by lilo
<code>lba32</code>	Allows addressing the disk above the 1024th cylinder
<code>install</code>	Selects menu presentation mode (text o menu)
<code>prompt</code>	Show a prompt (enable/disable install option)
<code>timeout</code>	Timeout in tenths of a second (150 = 15 sec)
<code>message</code>	A file containing the choices to be displayed
<code>default</code>	Default to boot after timeout, the word corresponds to a label

# Opciones por núcleo

➤ Opción	Descripción
<code>image</code>	Kernel image location
<code>initrd</code>	Initrd image location
<code>root</code>	Root for this kernel
<code>label</code>	Menu entry for this OS
<code>rootnoverify</code>	Como root para núcleos que GRUB no puede cargar
<code>chainloader</code>	Grub que pase el control a otro boot loader. Indicamos el sector donde se encuentra

# Instalación

- ★ Una vez definido el fichero de configuración podemos instalar **LILO** :

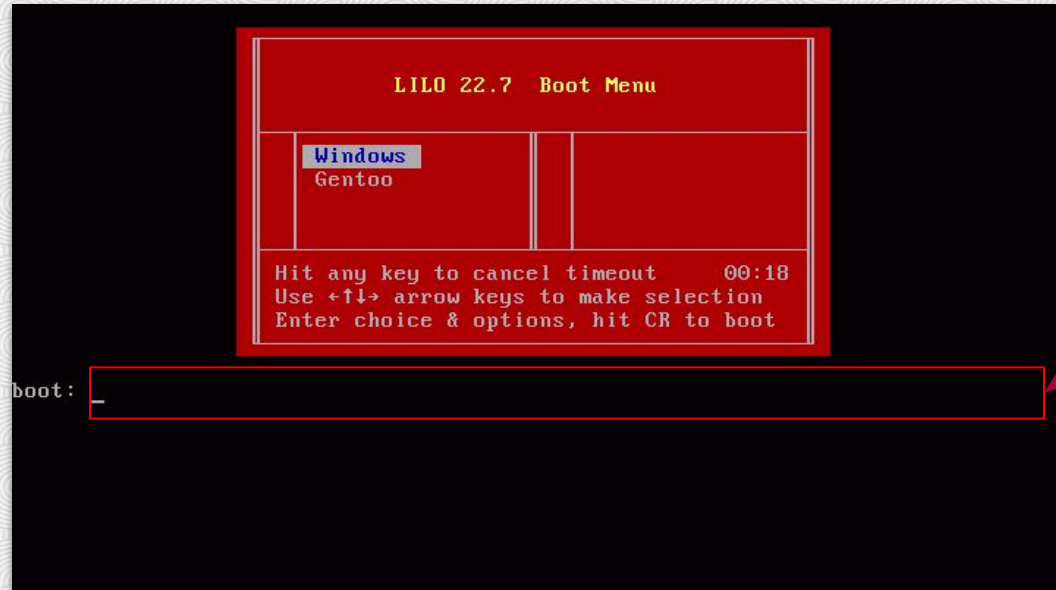
```
[root@localhost:] # /sbin/lilo -v -v
```

- ★ Podemos usar la opción **-C** para indicar la localización del fichero de configuración.
- ★ Se instalará donde indique el parámetro **boot**.
- ★ Cada vez que se modifica el fichero de configuración hay que volver a instalar LILO



# Arranque

- ★ Cuando arrancamos un equipo con LILO instalado veremos algo parecido a:



Cambiar opciones  
del núcleo

# Arranque en modo monousuario

- ★ Para arrancar el sistema en modo monousuario

```
boot: linux single
```

- ★ Para evitar que se puedan modificar las opciones de arranque:

```
password=clave  
restricted
```

- ★ Si queremos extender esta clave al arranque

```
password=clave  
restricted  
mandatory
```

# SysLinux

---

3

---

# Introducción

- ★ El proyecto SysLinux tiene como objetivo el desarrollo de cargadores de arranque livianos para labores específicas:
  - Para SF MS-DOS (**SYSLINUX**)
  - Para arranque desde la red (**PXELINUX**)
  - Para arranque desde cd-rom con la extensión "El Torito" (**ISOLINUX**)
  - Para SF ext2/ext3/ext4 o btrfs (**EXTLINUX**)
  - Para arranque de SO antiguos desde medios no tradicionales (**MEMDISK**)

# SYSLINUX

- ★ Es un cargador de arranque que se instala en sistemas de ficheros FAT (discos, disquetes)
- ★ Para instalar Syslinux desde linux en un SF de ficheros FAT, seguimos la sintaxis:

```
syslinux [options] <Device_Or_Image>
```

- ★ Por ejemplo, para instalarlo en un disquete

```
syslinux --install /dev/fd0
```

# Configuración

- ★ La configuración se realiza mediante el fichero de texto **syslinux.cfg** localizado en **/boot/syslinux/**, **/syslinux/** o **/**
- ★ Las rutas de los ficheros que aparecen en **syslinux.cfg** son relativas a la localización de **syslinux.cfg**:

```
$ vi /boot/syslinux/syslinux.cfg
DEFAULT linux
LABEL linux
    SAY Now booting the kernel from SYSLINUX...
    KERNEL vmlinuz.img
    APPEND ro root=/dev/sda1 initrd=initrd.img
```



# EXTLINUX

- ★ Similar a SYSLINUX pero para sistemas de ficheros ext2/ext3/ext4/Btrfs, XFS.
- ★ El sistemas de ficheros en el que se instalará debe estar montado.
- ★ El comando extlinux se encarga de instalar este bootloader

```
extlinux [options] <directory>
```

- ★ Por ejemplo, para instalarlo en un disquete:

```
mount /dev/fd0 /mnt  
extlinux --install /mnt
```

- ★ La partición donde reside la instalación debe estar marcada como de arranque



# Configuración

- ★ Se configura mediante el fichero **extlinux.conf** (3.x) o **syslinux.cfg** (4.x) que se encuentra en el mismo directorio donde se instaló **EXTLINUX**.
- ★ Las opciones son las mismas que en **SYSLINUX**

```
DEFAULT praxis-201043
PROMPT 1
TIMEOUT 60
LABEL praxis-201043
MENU LABEL Edoceo Praxis / 2010.43 / 3.6.34
KERNEL kernel-genkernel-x86_64-2.6.34-gentoo-r12-praxis
INITRD initramfs-genkernel-x86_64-2.6.34-gentoo-r12-praxis
APPEND root=/dev/sda4
```

# ISOLINUX

- ★ Es un bootloader para Linux/i386 que permite convertir en arrancables CD que siguen el estándar **ISO 9660/El Torito**.

**¡Atento!** A la hora de crear la imagen hay que tener en cuenta las restricciones del estándar

- ★ La extensión **IsoHybrid** permite crear USB autoarrancables.

# Instalación

- ★ Instalamos el paquete syslinux y mkisofs
- ★ Creamos el directorio **CD\_root**
- ★ Creamos el directorio **CD\_root\isolinux** y copiamos el fichero **isolinux.bin**
- ★ Creamos el fichero **isolinux.cfg** dentro de **CD\_root\isolinux**
- ★ Creamos el directorio **CD\_root/kernel** y copiamos el kernel a arrancar en este directorio
- ★ Ejecutamos

```
mkisofs -o output.iso -b isolinux/isolinux.bin -c isolinux/boot.cat \  
-no-emul-boot -boot-load-size 4 -boot-info-table CD_root  
cdrecord output.iso
```

# Configuración

```
display boot.txt
prompt 1
default 1
# Boot other devices
label a
    localboot 0x00
label b
    localboot 0x80
label c
    localboot -1
# PC-DOS
label 1
    kernel /kernel/memdisk
    append initrd=/images/tools.imz
```

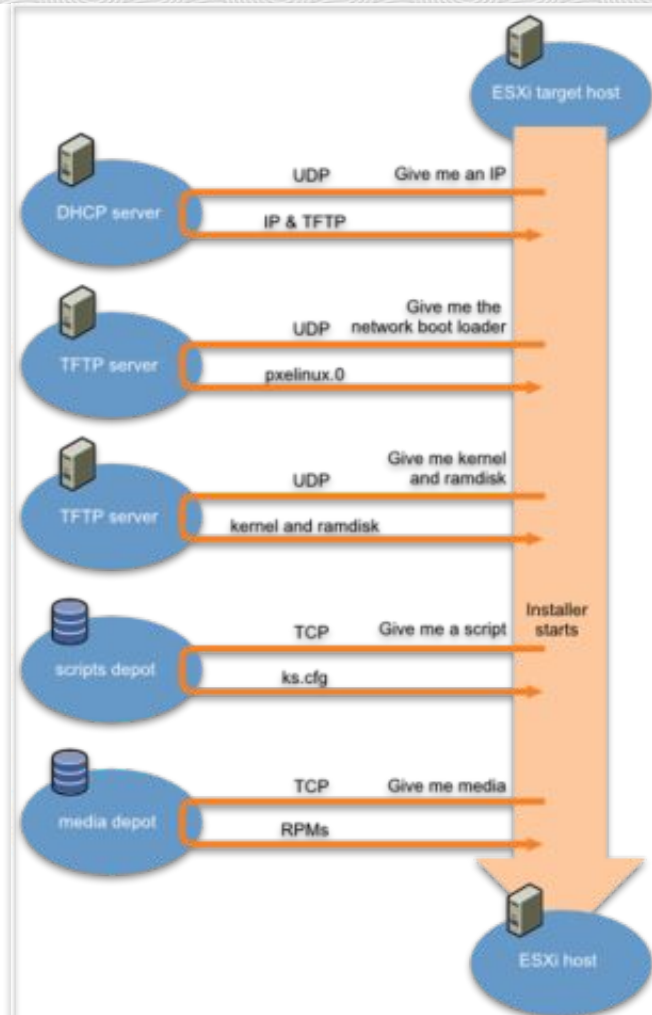
# Configuración

```
# Darik's Boot and Nuke
label 2
    kernel /kernel/memdisk
    append initrd=/images/bootnuke.imz

# MemTest
label 3
    kernel /kernel/memtp170
```

# PXELINUX

- ★ Permite el arranque de sistemas Linux a través de la especificación de un archivo de configuración (pxelinux.conf)



ando una ROM que sigue  
(ment)



# Configuración

- ★ Creamos el fichero `/tftpboot`
- ★ Copiamos los ficheros `pxelinux.0` (bootloader) y `ldlinux.c32` (módulos) a `/tftpboot`
- ★ Copiamos cualquier kernel y initrd que queramos arrancar en `/tftpboot`.
- ★ Creamos el directorio donde se almacenan los `pxelinux.cfg` que contendrán los ficheros de configuración para cada máquina.



## Configuración (II)

### ★ Ejemplo `pxelinux.cfg`:

```
default menu.c32
prompt 0
timeout 300
ONTIMEOUT local

MENU TITLE PXE Menu

LABEL Pmajic
    MENU LABEL Pmajic
    kernel images/pmagic/bzImage
    append noapic initrd=images/pmagic/initrd.gz root=/dev/ram0 init=/linuxrc
    ramdisk_size=100000
```

## Configuración (III)

### ★ Ejemplo `pxelinux.cfg`:

```
label Dos Bootdisk
    MENU LABEL ^Dos bootdisk
    kernel memdisk
    append initrd=images/622c.img

LABEL CentOS 5 x86 NO KS eth0
    MENU LABEL CentOS 5 x86 NO KS eth0
    KERNEL images/centos/5/x86/vmlinuz
    APPEND ks initrd=images/centos/5/x86_64/initrd.img ramdisk_size=100000
    ksdevice=eth1 ip=dhcp url --url http://xxx.xxx.xxx.xxx/mirrors/CentOS-5-i386/
```

## Aclaraciones

- ★ El servidor de DHCP mandará información sobre el proceso de arranque usando los siguientes valores no estándar

Opción	Valor	Descripción
208	pxelinux.magic	Desde PXELINUX 3.55, esta opción no se usa
209	pxelinux.configfile	Nombre del fichero de configuración PXELINUX
210	pxelinux.pathprefix	Directorio raíz donde se encuentra toda la configuración
211	pxelinux.reboottime	Segundo que transcurren antes de reiniciar el arranque normal en caso de fallo del TFTP.

## Aclaraciones (II)

- ★ Si no se encuentra el fichero indicado en `pxelinux.configfile`, se buscan teniendo en cuenta la mac del solicitante

# Otros bootloader

---

4

---

# Introducción

- ★ Existen multitud de cargadores de arranque
- ★ Muchos de ellos están pensados para arquitecturas concretas.
  - x86: podemos elegir entre GRUB o LILO (salvo sistemas AM64 que sólo admiten LILO)
  - Itanium Systems: usamos [ELILO](#)
  - IBM S/390 and IBM eServer zSeries Systems: usamos [z/IPL](#)
  - IBM eServer pSeries Systems: usamos [YABOOT](#)
- ★ Systemd tiene su propio gestor de arranque denominado [systemd-boot](#)



## U-boot

- ★ U-boot es un boot loader para varios tipos de arquitecturas de computadores, incluyendo PPC, ARM, AVR32, MIPS, x86, 68k, Nios, y MicroBlaze.



## Systemd-boot

- ★ [Systemd-boot](#) es un boot loader desarrollado dentro del proyecto systemd.
- ★ Se considera un UEFI boot manager capaz de ejecutar imágenes EFI.