

Arranque de una Computadora sin Disco Duro

Oscar Hernández Rivera y Luis G. de la Fraga

25 de Octubre 2002

Abstract

Este documento describe como configurar una computadora sin disco duro (diskless) en Linux.

1 Introducción

La idea central es que la computadora tenga un código de conexión o enlace de arranque en una memoria no volátil, por ejemplo un chip ROM, que permita conectar a un servidor y obtener un sistema de archivos sobre un enlace de red.

Los objetivos del trabajo son:

- Evitar el uso de discos duros para arrancar. Hay varias razones para hacer esto. Una es reducir el costo de mantenimiento de software sobre diferentes máquinas. Con el arranque sobre red los archivos están contenidos en un servidor central y pueden ser actualizados en un solo sitio.
- Otro objetivo es usar las computadoras en lugares donde los discos duros no tienen espacio suficiente.
- Otro objetivo es tener un sistema que pueda cambiar entre diferentes sistemas operativos sin tener que recargar el software.

Los nodos sin disco proveen movilidad para los usuarios, por ejemplo, el usuario puede entrar desde cualquier nodo sin disco sin estar en una sola estación de trabajo.

Los nodos sin disco pueden eliminar completamente la necesidad de unidades floppy, unidades de cd-rom, y disco duros.

Los nodos sin disco pueden tener solo una tarjeta de red, 32MB RAM, un CPU y una simple tarjeta madre la cual no tiene una interfaz de sockets/slots para discos duros, modem, cdrom, floppy, etc..

Con los nodos sin disco se pueden ejecutar programas sobre un Linux remoto.

Este proceso es útil para varias situaciones , por ejemplo:

1. Una terminal-X
2. Un cúmulo de computadoras
3. Ruteadores
4. Varios tipos de servidores remotos
5. Maquinas haciendo tareas en entornos hostiles hacia disco
6. Mantenimiento de software para cúmulos con configuración similar para estaciones de trabajo centralizadas.

2 Proceso de Booteo sobre Red

En orden para bootear sobre red, la computadora tiene que tener

1. Una identidad
2. Una imagen del sistema operativo y
3. usualmente, un sistema de archivos trabajando

La manera de distinguir una computadora de otra, es mediante una pieza de información que es única para la computadora, que es la dirección Ethernet.

Cada adaptador en el mundo tiene una dirección Ethernet única de 48 bits, porque cada fabricante de Hardware Ethernet les asigna un bloque de direcciones. Por convención estas direcciones están escritas en dígitos hexadecimales con dos puntos (:) separando cada grupo de 2 dígitos, por ejemplo:

```
00:60:08:C7:A3:D8.
```

El protocolo usado para obtener la dirección IP, dando una dirección Ethernet, es llamado boot protocol (BOOTP), pero también se puede usar Dynamic Host Configuration Protocol (DHCP). DHCP es una evolución de BOOTP.

Enseguida se muestra un ejemplo de intercambio de BOOTP:

Computadora sin disco: HOLA, mi dirección de hardware es 00:60:08:C7:A3:D8, por favor dame mi dirección IP.

Servidor BOOTP (Buscando dirección en base de datos): tu nombre es sigma1, tu dirección IP es **192.168.50.71**, tu servidor es **192.168.50.70**

Después de obtener una dirección IP, la computadora sin disco debe descargar una imagen del sistema operativo y ejecutarla.

Otro protocolo de internet usado aquí es el llamado Trivial File Transfer Protocol (TFTP).

En TFTP es como una reducción de la versión de FTP, pero aquí no hay autenticación, y este corre sobre protocolos de datagrama de usuario (User Datagram Protocol (UDP)) en lugar de el protocolo de control de transmisión (Transmission Control Protocol (TCP)). UDP fue escogido en lugar de TCP por simplicidad.

La implementación de UDP sobre una computadora sin disco puede ser pequeña, por lo tanto el código es fácil de acomodar en una ROM. Debido a que UDP es protocolo orientado a bloques, en contraposición a orientado a flujos, la transmisión va bloque por bloque parecido a esto:

Computadora: Dame el bloque 1 de /tftpboot/bootImage

TFTP server: Aquí está

Computadora: Dame el bloque 2

Y así sucesivamente, hasta que todo el archivo entero es transferido. El apretón de manos (handshaking) es un simple reconocimiento de cada esquema de bloques, y los paquetes perdidos son manejados mediante la retransmisión en intervalos. Cuando todos los bloques han sido recibidos, la ROM de booteo de red (BootROM) toma control sobre la imagen del sistema operativo como punto de entrada.

Finalmente, en orden para ejecutar un sistema operativo, un sistema de archivos raíz debe ser provisto. El protocolo usado por Linux y otros Unix es normalmente el sistema de archivos de red (NFS), aunque otras opciones son posibles. En este caso el código no tiene que alojarse en la ROM, pero puede ser parte del sistema operativo que se descargó. Sin embargo el sistema operativo debe ser capaz de ejecutarse con un sistema de archivos raíz que esta montado por NFS, en lugar de un disco real.

3 Creación la Imagen de Arranque

Primero se tiene que conocer que tipo de tarjeta de red se tiene instalada en el nodo sin disco, esto se puede saber checando la configuración del setup del bios, o si se tiene la tarjeta instalada en un sistema Linux mediante el archivo modules.conf que se encuentra en /etc, ó mediante la herramienta lspci que se encuentra en /sbin del paquete pciutils, que se puede descargar de rpmfind.net; por ejemplo:

```
[root@sigma]$cat /etc/modules.conf
[root@sigma]$/sbin/lspci
```

los cuales nos muestran algo similar en el caso del modules.conf

```
alias eth0 8139too
alias parport_lowlevel parport_pc
```

y en el caso de lspci

```
00:07.0 Ethernet controller: Realtek Semiconductor Co., Ltd. RTL-8139/8139C (rev 10)
00:00.0 Host bridge: VIA Technologies, Inc. VT8367 [KT266]
00:01.0 PCI bridge: VIA Technologies, Inc. VT8367 [KT266 AGP]
```

para este caso se utilizo una tarjeta de red tipo rtl8139

después de conocer que tipo de tarjeta tenemos, se puede instalar alguna de las dos distribuciones gratuitas "etherboot" (etherboot.sourceforge.net) o "Netboot" (www.netboot.com) y para generar la bootrom que utilizaremos en el floppy de arranque, pero en este caso se genero mediante una página WEB en www.rom-o-matic.com en donde se pueden configurar algunas opciones de arranque, después que se tiene la bootrom, se hace un cat a la bootrom direccionando la salida hacia un floppy

Ejemplo:

```
[root@sigma]$cat eb-5.0.6-rtl8139.lzdsk > /dev/fd0
```

Pero antes de poner el floppy de arranque para probar el booteo, tenemos que configurar 4 servicios sobre el servidor de los cuales bootp y tftp son administrados con xinetd.

1. Xinetd
2. Bootp (o DHCP)
3. Tftp
4. Nfs

No se tiene que configurar todos al mismo tiempo , se pueden hacer paso por paso, verificando que cada uno funcione antes de ir al siguiente.

3.1 Instalando xinetd

El rpm del xinetd fue instalado del CD-ROM de Linux RedHat 7.3. Ejemplo:

```
[root@sigma]$ rpm -i xinetd-2.3.4-0.8.i386.rpm
[root@sigma]$ rpm -Uvh xinetd-2.3.4-0.8.i386.rpm
```

las opciones -Uvh se usan para hacer una actualización del programa xinetd.

3.2 Instalando bootp

En este caso se instalo la vesion 2.4.3-7 del Bootp que se descargo de rmpfind.net, ejemplo:

```
[root@sigma]$ rpm -i bootp-2.4.3-7.i386.rpm
[root@sigma]$ rpm -Uvh bootp-2.4.3-7.i386.rpm
```

por si se tiene una versión anterior instalada en el sistema.

3.3 Instalando tftp

En este caso también se descargo la versión mas reciente 0.17-16 del rpm del tftp de rmpfind.net. Ejemplo:

```
[root@sigma]$ rpm -i tftpd-0.17-16.i686.rpm
[root@sigma]$ rpm -Uvh tftpd-0.17-16.i686.rpm
```

Para ver si cada uno de estos paquetes se instalaron correctamente en el sistema podemos hacerlo siguiente:

```
[root@sigma]$ rpm -qa > sale
[root@sigma]$ grep xinetd* sale
[root@sigma]$ grep boot* sale
[root@sigma]$ grep tftp*
```

Ya que se tiene instalados estos servicios, se procede a configurarlos. Para ello se usa xinetd el cual lee un archivo de configuración para cada servicio, el cual después tiene que reiniciarse para poder habilitar a los servicios (bootp y tftp), los archivos de configuración son situados en /etc/xinetd.d se pueden crear con vi o cualquier editor de su preferencia

Ejemplo de configuración del demonio bootps:

```
[root@sigma]$ vi bootps
```

el cual tiene el siguiente contenido

```
service bootps
{
    socket_type      =      dgram
    wait            =      yes
    user            =      root
    server          =      /usr/sbin/bootpd
    log_on_success  +=     HOST PID
    log_on_failure  +=     HOST RECORD
    protocol        =      udp
    server_args     =      -t 5 -d 5
}
}
```

después se guarda y se mueve a /etc/xinetd.d

Ejemplo de configuración del demonio tftp:

```
[root@sigma]$ vi /etc/xinetd.d/tftp
```

el cual tiene el siguiente contenido

```
service tftp
{
    socket_type      =      dgram
    wait            =      yes
    user            =      root
    server          =      /usr/etc/in.tftpd
    server_args     =      -s /tftpboot
}
}
```

Para ver si están activados los servicios de bootp y tftp ponemos la siguiente sentencia

```
[root@sigma]$ /sbin/chkconfig --list
```

el cual nos muestra los servicios agregados para xinetd

```
xinetd          0:off  1:off  2:off  3:on   4:on   5:on   6:off
servicios basados en xinetd:
    tftp:        on
    bootps:      on
```

Configuración del archivo bootptab

Después que son agregados los dos servicios el demonio bootpd, lee un archivo de configuración llamado bootptab, en el cual se encuentran sentencias necesarias para cada máquina sin disco; este archivo está localizado en /etc

Ejemplo de bootptab:

```
.default:\
:hd=/tftpboot:bf=bootImage:\
:sm=255.255.255.0:\
:vm=auto:\
```

```
:ds=148.247.2.1:\
:gw=192.168.50.1:\
:sa=192.168.50.70:
```

```
sigma1:ht=ethernet:ha=00308488d674:rp=/tftpboot/sigma1:ip=192.168.50.71:tc=.default:
```

Si queremos agregar mas de una computadora sin disco, ponemos una sentencia para cada una:

```
sigma2:ht=ethernet:ha=00308488d3f5:rp=/tftpboot/sigma2:ip=192.168.50.72:tc=.default:
sigma3:ht=ethernet:ha=00308487e3cb:rp=/tftpboot/sigma3:ip=192.168.50.73:tc=.default:
sigma4:ht=ethernet:ha=00308488e0a1:rp=/tftpboot/sigma4:ip=192.168.50.74:tc=.default:
sigma5:ht=ethernet:ha=00308487edb1:rp=/tftpboot/sigma5:ip=192.168.50.75:tc=.default:
sigma6:ht=ethernet:ha=0030848902ba:rp=/tftpboot/sigma6:ip=192.168.50.76:tc=.default:
sigma7:ht=ethernet:ha=003084890866:rp=/tftpboot/sigma7:ip=192.168.50.77:tc=.default:
```

Estas opciones son tomadas por defecto para cada máquina. Cada campo significa:

hd contiene el directorio home del archivo de booteo (/tftpboot)

bf contiene el nombre del archivo de booteo (bootImage)

sm contiene la mascara de subred (255.255.255.0)

vm contiene el seleccionador de proveedor de cookies (auto)

ds contiene la lista de direcciones del nombre de dominio del servidor (148.247.2.1)

gw contiene la dirección del nombre de dominio del servidor (Gateway)

sa contiene la dirección del servicio TFTP que el cliente debería usar (192.168.50.70)

Después de esto, cada maquina debe tener una linea, en donde cada campo es separado por dos puntos (:)

```
sigma1:ht=ethernet:ha=00308488d674:rp=/tftpboot/sigma1:ip=192.168.50.71:tc=.default:
```

donde cada campo significa:

El primer campo contiene el nombre del host (sigma1).

ht contiene el tipo de hardware de la computadora sin disco (ethernet)

ha contiene la dirección de la tarjeta ethernet (00308488d674)

rp contiene la ruta de montaje del sistema de archivos (/tftpboot/sigma1)

ip contiene la dirección IP de la computadora sin disco (192.168.50.71)

tc contiene las opciones por defecto, que fueron agregas anteriormente (.default)

Para poder saber que dirección tiene la tarjeta de red, arrancamos la computadora sin disco con el floppy que creamos, la computadora carga la bootrom en la RAM e inicia el proceso de arranque en donde nos muestra la dirección en forma hexadecimal, separando 2 dígitos con dos puntos (:) este numero es el que escribimos en el campo de **ha** pero sin los 2 puntos

Después reiniciamos el servicio de xinetd para que pueda leer el bootptab mediante la siguiente sentencia

```
[root@sigma]$ /etc/rc.d/init.d/xinetd restart
```

Ahora arrancamos la computadora sin disco con el floppy y este debería detectar la tarjeta Ethernet y emitir una petición hacia BOOTP. Si todo va bien, el servidor debería responder a la computadora sin disco con la información requerida.

Desde que /tftpboot/bootImage no existe, se suspenderá cuando trate de cargar este archivo.

Ahora se necesita compilar un kernel personalizado, es decir habilitar algunas opciones del kernel para poder montar el sistema de archivos y los drivers de la tarjeta ethernet, etc.

4 Instalación del Kernel Personalizado

Para la compilación se utilizó el rpm del código fuente del kernel (kernel-source-2.4.18-3.i386.rpm) del CD-ROM de Linux RedHat 7.3, que es la misma versión del kernel del sistema en donde se compiló, pero también se puede descargar del sitio oficial de redhat (ftp.redhat.com en /pub/redhat/redhat-7.3-en/os/i386/RedHat/RPMS), también se realizaron pruebas con diferentes versiones de kernel "linux-2.4.9.tar.gz" descargada de ftp.kernel.org en el directorio /pub/linux/kernel/v2.4 y el kernel "linux-2.5.9.tar.gz" del directorio v2.5, pero había incompatibilidad con la carga de módulos a la hora de bootear el sistema .

Para poder instalar el kernel se realizó lo siguiente

```
[root@sigma]$ rpm -i kernel-source-2.4.18-3.i386.rpm
```

el cual se instala en el directorio /usr/src, en donde crea un directorio llamado linux-2.4.18-3 nos cambiamos a ese directorio, en donde podemos ejecutar el programa configurador del kernel, el cual es una interface para poder habilitar las opciones que se necesitan, las opciones que existen son:

```
[root@sigma]$ make xconfig
```

Esta invocará la versión gráfica de la utilidad de configuración del kernel.

```
[root@sigma]$ make menuconfig
```

Esta invocará la versión basada en menus (muy útil también).

```
[root@sigma]$ make config
```

Esta invocará la versión más simple en modo texto basada en línea-por-línea (solo para expertos).

Configuración del kernel

En la sección de block devices habilitamos la opción de LoopBack device support ,también habilitamos la opción de Network block device support, como se muestra en la figura.

Después en la sección de Enter Device Support -¿ Ethernet 10 or 100 Mbit habilitamos la tarjeta de red que tengamos o similar como se muestra en la figura.

Después que habilitamos la tarjeta de red , habilitamos la opción de IP (point-to-point protocol) support

Después pasamos a la sección de Networking options y habilitamos la opción de "IP:Kernel level autoconfiguration", también seleccionamos "BOOTP support" ò "DHCP support"dependiendo de que se está utilizando, el "RARP support" no es necesario.

Después en la sección de File system habilitamos las opciones de kernel automounter support y Kernel automounter version 4 support (also support v3) y más hacia abajo habilitamos las opciones de /proc file system support, /dev file system support (EXPERIMENTAL) y Automatically mount at boot, como se muestra en la figura.

Después en la parte inferior pasamos a la subsección de Network File System en donde vamos a habilitar la opción de NFS file system support, Root file system on NFS, NFS server support y Provide NFSv3 server support, como se muestra en la figura.

Después de realizar la configuración del kernel se guardan los cambios y se sale.

Compilación del kernel

La compilación del kernel es muy sencilla se hace mediante la siguiente línea:

```
[root@sigma]$ make dep && make clean && make bzImage
```

Las opciones dep y clean son propias del kernel y la bzImage es la imagen que vamos a necesitar para poder arrancar Linux en la computadora sin disco.

En este momento se puede ir a tomar un café o hacer cualquier cosa por que este proceso tarda varios minutos, después que se termine el proceso de compilación se genera la bzImage en el directorio /usr/src/linux-2.4/arch/i386/boot/, la cual se tiene que convertir dentro de una imagen etiquetada.

Nota: si al finalizar la compilación del la imagen del kernel nos aparece un mensaje de que el tamaño del kernel es demasiado grande para poder estar en un floppy se pueden deshabilitar varias opciones del kernel (scsi y PCMCIA, o cualquier otra cosa que no se requiera) para poder reducir el espacio en la bootimage.

Una imagen etiquetada es una imagen de kernel normal con una cabecera especial que le dice al bootloader de red, donde van los bytes en memoria y en que dirección inicia el programa.

Esta etiquetación es realizado con el programa "mknbi-linux" el cual se descarga e instala de rpmfind.net, o también se encuentra en el paquete etherboot o netboot si es que se instalaron.

Ejemplo:

Block devices

Block devices

<input type="checkbox"/> y	<input type="checkbox"/> m	<input type="checkbox"/> n	OnSpec 90c26 protocol	Help
<input type="checkbox"/> y	<input type="checkbox"/> m	<input checked="" type="checkbox"/> n	Compaq SMART2 support	Help
<input type="checkbox"/> y	<input type="checkbox"/> m	<input checked="" type="checkbox"/> n	Compaq Smart Array 5xxx support	Help
<input type="checkbox"/> y	<input type="checkbox"/> -	<input type="checkbox"/> n	SCSI tape drive support for Smart Array 5xxx	Help
<input type="checkbox"/> y	<input type="checkbox"/> m	<input checked="" type="checkbox"/> n	Mylex DAC960/DAC1100 PCI RAID Controller support	Help
<input checked="" type="checkbox"/> y	<input type="checkbox"/> m	<input type="checkbox"/> n	Loopback device support	Help
<input checked="" type="checkbox"/> y	<input type="checkbox"/> m	<input type="checkbox"/> n	Network block device support	Help
<input type="checkbox"/> y	<input type="checkbox"/> m	<input checked="" type="checkbox"/> n	RAM disk support	Help
4096			Default RAM disk size	Help
<input type="checkbox"/> y	<input type="checkbox"/> -	<input type="checkbox"/> n	Initial RAM disk (initrd) support	Help

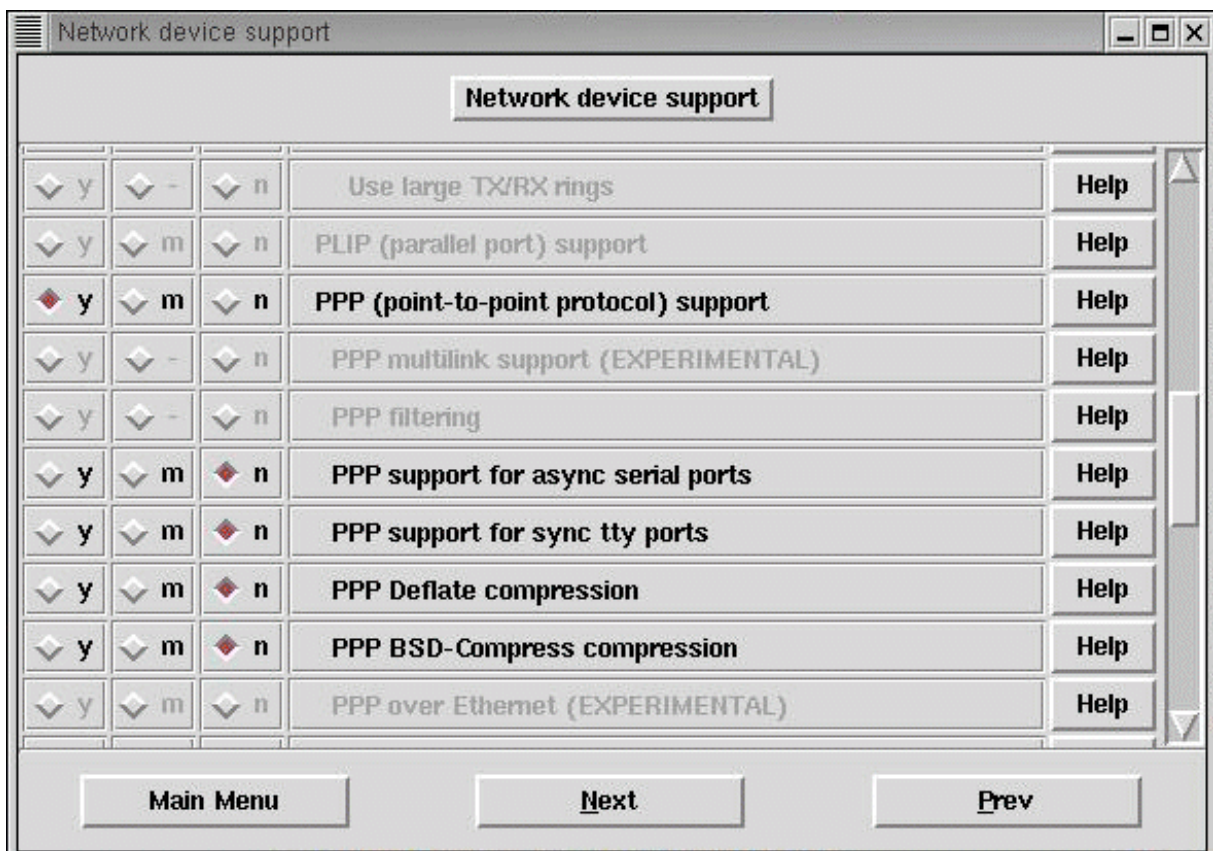
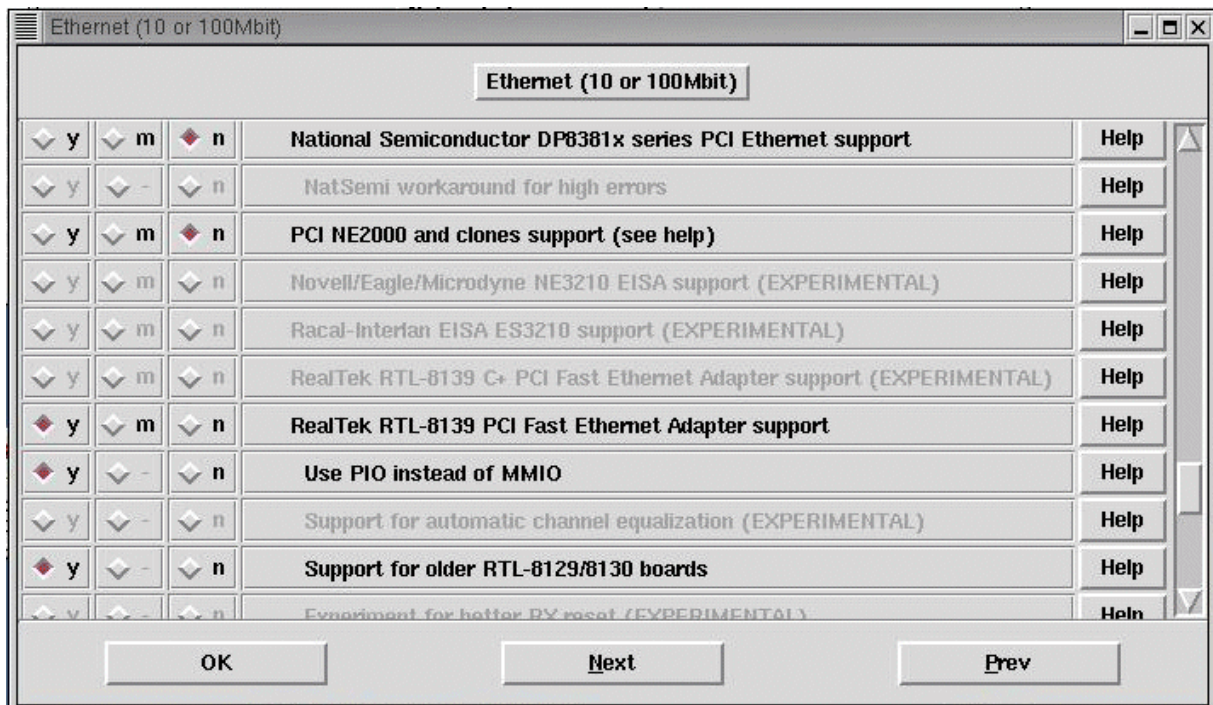
Main Menu Next Prev

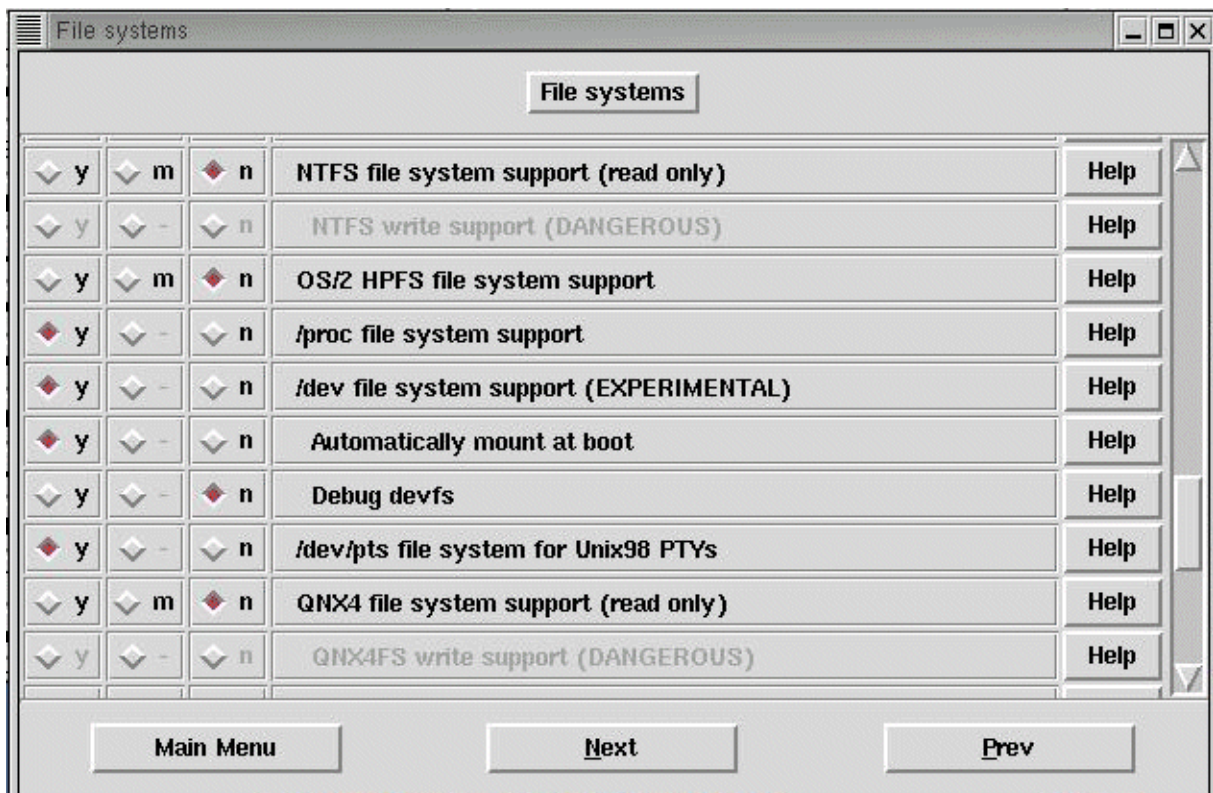
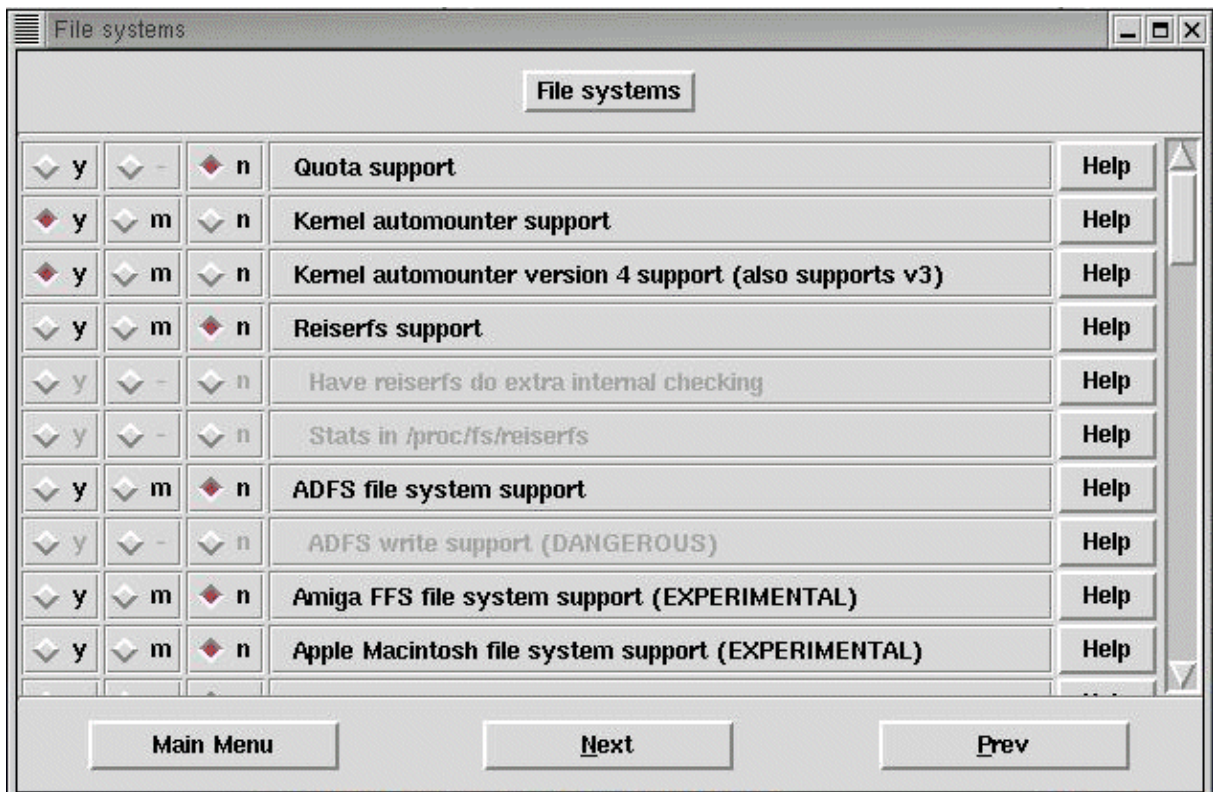
Network device support

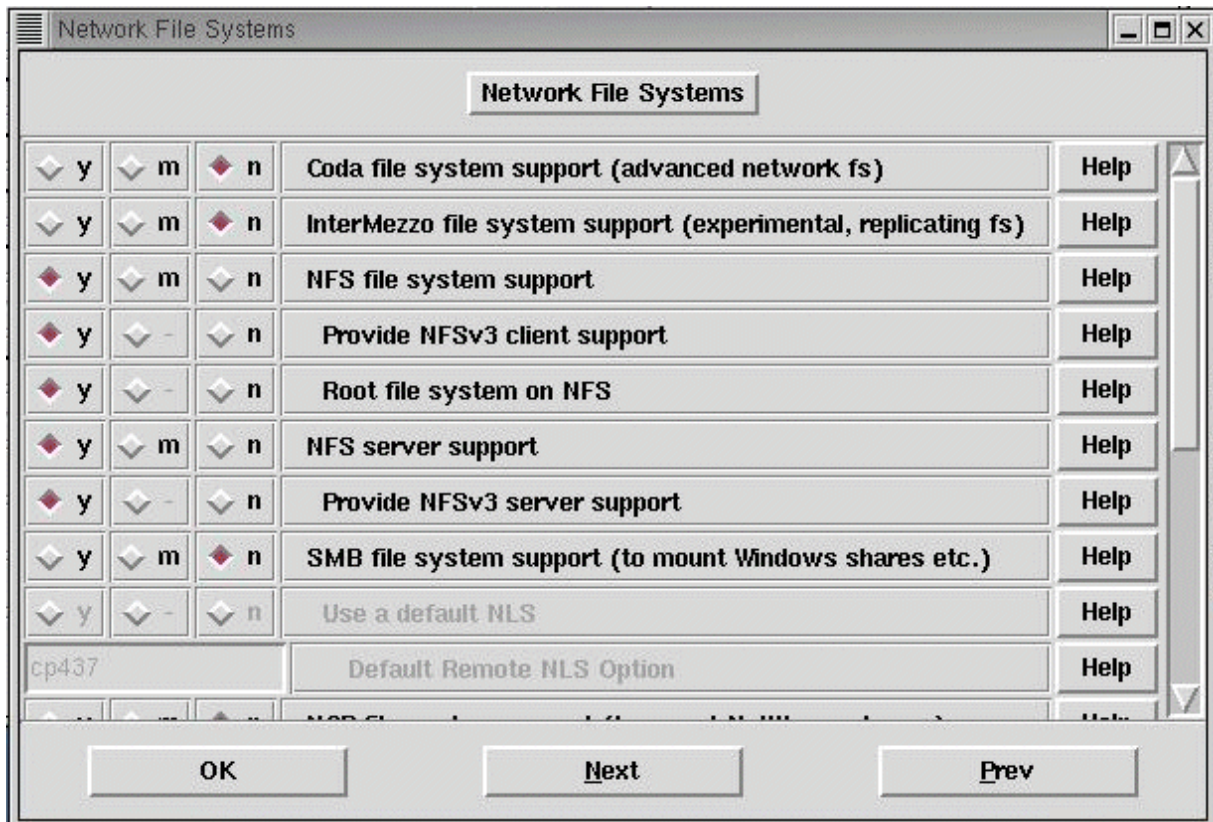
Network device support

<input type="checkbox"/> y	<input type="checkbox"/> m	<input checked="" type="checkbox"/> n	EQL (serial line load balancing) support	Help
<input type="checkbox"/> y	<input type="checkbox"/> m	<input checked="" type="checkbox"/> n	Universal TUN/TAP device driver support	Help
<input type="checkbox"/> y	<input type="checkbox"/> m	<input type="checkbox"/> n	Ethertap network tap (OBSOLETE)	Help
<input type="checkbox"/> y	<input type="checkbox"/> m	<input checked="" type="checkbox"/> n	General Instruments Surfboard 1000	Help
			Ethernet (10 or 100Mbit)	
			Ethernet (1000 Mbit)	
<input type="checkbox"/> y	<input type="checkbox"/> -	<input checked="" type="checkbox"/> n	FDDI driver support	Help
<input type="checkbox"/> y	<input type="checkbox"/> m	<input type="checkbox"/> n	Digital DEFEA and DEFPA adapter support	Help
<input type="checkbox"/> y	<input type="checkbox"/> m	<input type="checkbox"/> n	SysKonnnect FDDI PCI support	Help
<input type="checkbox"/> y	<input type="checkbox"/> -	<input type="checkbox"/> n	HIPPI driver support (EXPERIMENTAL)	Help

Main Menu Next Prev







```
[root@sigma]$ rpm -Uvh mknbi-1.2-6.noarch.rpm
```

la etiquetacion se realiza mediante la siguiente linea:

```
[root@sigma]$ /sbin/mknbi-linux -d rom -i rom -k bzImage -o bootImage
```

después se mueve la bootImage resultante hacia donde esta el directorio asignado para las computadoras sin disco, que en este caso es el /tftboot/.

Nota: el nombre de bootImage es el nombre que esta el el archivo de configuración /etc/bootptab , y puede ser el nombre que se desee.

5 Configurando el Servidor NFS

Sistema de archivos de red (NFS) fue desarrollado para permitir montar particiones de disco sobre una maquina remota como si estuviera sobre un disco local. Esto permite compartir archivos a través de la red de una manera muy rápida.

También nos da el potencial de negar el acceso a gente no permitida hacia el disco duro sobre la red, leer correos y borrar archivos.

Esto es necesario para poder hacer el sistema de archivos para cada computadora sin disco, montarlos y tener acceso individualmente.

Se requiere tener instalados nfs-utils y portmap. Preguntaremos al sistema si estos están instalados con la siguiente línea de comando:

```
[root@sigma]$ rpm -q nfs-utils portmap
```

Lo cual debe de regresarle algo como lo siguiente:

```
nfs-utils-0.3.3-5
portmap-4.0-41
```

pero si no se tienen se deberán instalar, en este ejemplo se utilizo el rpm `nfs-utils-0.3.3-5.i386.rpm` `portmap-4.0-41.i386.rpm` descargados de `rpmfind.net`, pero también se pueden instalar del CD-ROM de Linux RedHat 7.3 ejemplo:

```
[root@sigma]$ rpm -Uvh nfs-utils-0.3.3-5.i386.rpm
[root@sigma]$ rpm -Uvh portmap-4.0-41.i386.rpm
```

Ya instalados los paquetes, se procede a realizar la configuración de los archivos `/etc/exports`, `/etc/hosts.allow` y `/etc/hosts.deny`

Configurando /etc/exports

Este archivo contiene una lista de entradas; cada entrada indica un volumen que es compartido y como es compartido. Se puede checar el manual (`man exports`) para la completa descripción de todas las opciones para los archivos.

una entrada en `/etc/exports` típicamente es parecida a esto :

`directorio maquina1(opcion1,opcion2) maquina2(opcion1,opcion2)`

Donde `directorio` es el directorio que queremos compartir, si se comparte un directorio, entonces todos los subdirectorios serán compartidos también

La `maquina1` y `maquina2` son maquinas clientes (sin disco) que tendrán acceso a los directorios. Las maquinas pueden ser listadas por sus direcciones de DNS y sus direcciones IP (ejemplo: `maquina1.lab.com` o `192.168.0.8`). Para este caso se utilizo el nombre de `sigma1` para la maquina

Hay una lista de opciones para cada maquina que describirá que tipo de acceso tendrá la maquina. Las opciones mas importante son:

`ro`: el directorio es compartido como solo lectura; el cliente no podrá escribir en este directorio. Esto opción es por defecto.

`rw`: la maquina cliente (sin disco) tendrá acceso para leer y escribir en el directorio.

`no_root_squash`: por defecto, cualquier pedido de archivo hecho por el usuario `root` sobre la maquina cliente es tratado como si fuera hecho por el usuario `nobody` en el servidor. Si no es seleccionado `no_root_squash`, entonces `root` sobre la maquina cliente tendra el mismo nivel de acceso a los archivos del sistema, como si se localizara en el sistema de archivos del servidor.

El ejemplo del archivo `/etc/exports` utilizado para este ejemplo:

```
/tftpboot/sigma1  sigma1(rw,no_root_squash)
/usr              sigma1(ro,no_root_squash)
/usr/local        sigma1(ro,no_root_squash)
/home             sigma1(rw,no_root_squash)
```

El todos los directorios mostrados en este ejemplo son exportados hacia `sigma1` que es la maquina cliente.

`/tftpboot/sigma1` es en donde se encuentra el sistema de archivos para cada computadora sin disco en `/usr` y `/usr/local` contienen los programas y comandos para la computadora, `/home` contiene las cuentas de usuarios.

*Nota: si se desean poner mas de una computadora sin disco se debe agregar cada directorio para cada una.

Después se configuran el archivo `/etc/hosts.allow` y el `/etc/hosts.deny`

Estos dos archivos especifican cual de las computadoras sobre la red pueden usar servicios sobre la maquina. Cada linea del archivo contiene una simple lista de entradas de servicios y un conjunto de maquinas.

ejemplo:

```
portmap: 192.168.50.0/255.255.255.0
```

para la versión reciente de `nfs-utils`, tendremos que agregar lo siguiente.

```
lockd: 192.168.50.71
rquotad: 192.168.50.71
mountd: 192.168.50.71
statd: 192.168.50.71
```

Ya que se definieron los volúmenes a compartir, solo resta iniciar o reiniciar el servicio `nfs`. Utilice cualquiera de las dos líneas dependiendo el caso:

```
[root@sigma]$ /etc/rc.d/init.d/nfs stop
[root@sigma]$ /etc/rc.d/init.d/nfs start
```

ó solamente

```
[root@sigma]$ /etc/rc.d/init.d/nfs restart
```

A fin de asegurarnos de que el servicio de nfs esté habilitado la siguiente vez que se encienda el equipo, debemos ejecutar lo siguiente:

```
/sbin/chkconfig --level 345 nfs on
```

El comando anterior hace que se habilite NFS en los niveles de corrida 3, 4 y 5.

Como medida de seguridad adicional, si tiene un cortafuegos implementado, cierre, para todo aquello que no sea parte de su red local, los puertos tcp y udp 2049, ya que estos son utilizados por NFS para escuchar peticiones.

Para verificar si están corriendo los servicios de portmap y nfs ejecutamos la siguiente línea:

```
[root@sigma]$ rpcinfo -p
```

El cual nos va a mostrar algo similar a esto

```
program vers proto  port
 100000    2    tcp    111  portmapper
 100000    2    udp    111  portmapper
 100011    1    udp    749  rquotad
 100011    2    udp    749  rquotad
 100005    1    udp    759  mountd
 100005    1    tcp    761  mountd
 100005    2    udp    764  mountd
 100005    2    tcp    766  mountd
 100005    3    udp    769  mountd
 100005    3    tcp    771  mountd
 100003    2    udp    2049 nfs
 100003    3    udp    2049 nfs
 300019    1    tcp    830  amd
 300019    1    udp    831  amd
 100024    1    udp    944  status
 100024    1    tcp    946  status
 100021    1    udp    1042 nlockmgr
 100021    3    udp    1042 nlockmgr
 100021    4    udp    1042 nlockmgr
 100021    1    tcp    1629 nlockmgr
 100021    3    tcp    1629 nlockmgr
 100021    4    tcp    1629 nlockmgr
```

Con esto nos aseguramos que se están ejecutando los servicios necesarios para montar el sistema de archivos y los directorios necesarios para el buen funcionamiento.

6 Configurando los Clientes

La computadora sin disco pide montar /tftpboot/;nombre de la computadora en bootptab; (tftpboot/sigma1) como su directorio raíz / por medio de NFS desde el servidor. La / (raíz) debe contener /sbin, /bin, /lib, /etc, /var, /tmp, /root, /dev and /usr, /proc.

Los directorios /sbin, /dev, /bin, /lib/, /etc, puede ser una copia del sistema de archivos del servidor. los cuales pueden ser compartidos para todas las computadoras sin disco que se deseentener.

Personalizar el /etc/sysconfig/network, /etc/sysconfig/network-scripts/ifcfg-eth0, /etc/fstab, /etc/modules.conf, para cada máquina. Quitar todos los servicios que no se necesiten. Remover todas las cosas que no se necesitan de /var, ejemplo la base de datos de los RPM's, archivos lpd. /root and /proc solo deben de existir. /tmp también solo debe de existir en modo 777.

para este caso de montan los directorios /usr y /home para minimizar el espacio para cada computadora sin disco en el servidor, /usr se monta con permisos de solo lectura (ro).

La imagen del kernel también puede ser compartida (/tftpboot/bootImage) para cada computadora sin disco.

para este ejemplo se comprimieron los directorios /etc, /lib /dev mediante la siguiente sentencia:

```
[root@sigma]$ tar zcvf etc.tar /etc
[root@sigma]$ tar zcvf lib.tar /lib
[root@sigma]$ tar zcvf dev.tar /dev
```

y se descomprimieron en la raíz de cada TDC (/tftpboot/sigma1....n) mediante la siguiente línea :

```
[root@sigma]$ tar zvxf etc.tar .
[root@sigma]$ tar zvxf lib.tar .
[root@sigma]$ tar zvxf dev.tar .
```

después se configura el /etc/fstab, /etc/sysconfig/network, /etc/sysconfig/network-scripts/ifcfg-eth0 y /etc/modules.conf
Ejemplo del archivo /etc/fstab:

```
sigma:/tftpboot/sigma1 / nfs defaults 1 1
sigma:/usr /usr nfs defaults 1 1
sigma:/home /home nfs defaults 1 1
sigma:/usr/local /usr/local nfs defaults 1 1
/dev/cdrom /mnt/cdrom iso9660 noauto,owner,kudzu,ro 0 0
/dev/fd0 /mnt/floppy auto noauto,owner,kudzu 0 0
none /dev/pts devpts gid=5,mode=620 0 0
none /proc proc defaults 0 0
```

ejemplo del /etc/sysconfig/network:

```
NETWORKING=yes
HOSTNAME=sigma1
GATEWAY=192.168.50.1
```

ejemplo del /etc/sysconfig/network-scripts/ifcfg-eth0:

```
DEVICE=eth0
BOOTPROTO=static
BROADCAST=192.168.50.255
IPADDR=192.168.50.71
NETMASK=255.255.255.0
NETWORK=192.168.50.0
ONBOOT=yes
```

ejemplo del /etc/modules.conf

```
alias parport_lowlevel parport_pc
alias eth0 8139too
```

Estos ejemplos de configuraciones de archivos son para cada computadora que se desee y también dependen del tipo de dispositivos que se tengan y de la red que utilice.

Ya que se tiene todo configurado se arranca la máquina sin disco con el floppy, la cual empezará a recibir todos los datos necesarios para poder arrancar y pondrá la imagen del kernel en memoria para poder iniciar el sistema tal y como se inicia normalmente.

Cualquier duda se pueden consultar las siguientes referencias.

References

- [1] <http://etherboot.sourceforge.net>
- [2] <http://kb.redhat.com/view.php?eid=185>
- [3] <http://www.ibiblio.org/mdw/HOWTO/mini/NFS-Root.html>
- [4] <http://www.redhat.com/mirrors/LDP/HOWTO/Diskless-HOWTO.html>
- [5] <http://netboot.sourceforge.net/english/index.shtml>
- [6] <http://nfs.sourceforge.net/>
- [7] <http://www.cisco.com/univercd/cc/td/doc/product/iaabu/cddm/cddm111/adguide/dhcp.htm>