Linux sin disco Mini-COMO

Robert Nemkin, buci@math.klte.hu
Traducido por Rodolfo Pilas, rodolfo@linux.org.uy v0.0.3 12 Sep 1996

Este documento describe cómo configurar un sistema Linux sin disco.

Índice General

1 Copyright

 Deseo agradecer a Bela Kis `bkis@cartan.math.klte.hu` por traducir este documento al inglés.

2 Cambios

- v0.0.3 12 Sept 1996: Corrección de pequeños errores.

3 Cómo configurar una máquina Linux sin disco

Este documento trata sobre cómo configurar una máquina Linux sin disco. Algunas veces puede ser necesario correr Linux en PCs que no tienen ni discos ni disqueteras. Esto es posible, por ejemplo, en red con otro sistema Unix corriendo `bootp`, `tftp` y servidor NFS, y una EPROM (burner) en las máquinas Linux sin disco/disquetera.

4 Documentación de referencia.

- Raíz-NFS Mini COMO
- Linux NET-2/3-HOWTO de Terry Dawson, `94004531@postoffice.csu.edu.au`
- `/usr/src/linux/README` referido a la configuración y compilación de nuevos núcleos.

5 Hardware

Para preparar este documento el sistema fue chequeado utilizando la siguiente configuración:

- Sun-OS 4.1.3 como servidor de arranque
- Slackware 2.3 + Linux 1.2.8 + Tarjeta ethernet Western Digital 8013
- Red Ethernet

6 Conceptos fundamentales

La idea fundamental es la siguiente: el PC buscará su dirección IP en el servidor de arranque vía protocolo `bootp`, usando `0.0.0.0` como dirección IP inicial y recibiendo su núcleo vía el protocolo `tftp`. Arrancar a través de segmentos (vía un *router* o encaminador) no es simple: se puede o bien colocar ambos, servidor y las computadoras sin disco, en el mismo segmento de red o configurar un asistente de direcciones UDP en su encaminador hacia la dirección del servidor. Refiérase al manual de su encaminador para obtener más información.

Para hacer esto siga los siguientes pasos:

6.1 Configuración del PC

Instale el paquete `nfsboot` (el paquete estará disponible en su servidor Linux favorito en el directorio `/pub/Linux/system/Linux-boot`). Este paquete contiene la imagen de una EPROM de arranque para la tarjeta wd8013 que puede ser grabada directamente.

Existen varias vías alternativas para preparar el PC:

- si la máquina no está totalmente exenta de discos, entonces es posible utilizar un pequeño programa DOS, o
- la imagen binaria del disquete contenida en el mismo paquete. Si elije esta última opción puede escribir la imagen en el disquete con el comando `dd`:

```
dd if=imagen of=/dev/fd0H1440
```

Estas imágenes contienen un cliente `bootp` y `tftp`. Es necesario preparar también un núcleo de Linux, el cual tenga la opción `nfs-root` habilitada.

- Si está usando un núcleo 1.2.13, es necesario poner al día el núcleo con el parche incluído en el paquete `nfsboot`. Refiérase a `patch(1)`
- Si está usando uno de los últimos núcleos (1.3.x o superior) tiene que habilitar la opción `nfs-root`.

Puede o no configurar soporte para dispositivos de sistemas de archivo (disquetes o discos duros), pero es obligatorio habilitar soporte TCP/IP, soporte para tarjeta ethernet y soporte de sistema de archivos nfs. Y, una vez realizado esto, recompilar el núcleo de la manera acostumbrada.

6.2 Instalación de `bootpd` en el servidor

Bootpd puede encontrarse en el paquete `bootpd-2.4.tar.gz` (el cual se obtiene de su servidor favorito de software Linux en el directorio `/pub/Linux/system/Network/boot.net`). Obtenga el paquete, compílelo e instálelo. Si alguna máquina Linux tiene la distribución Slackware, es posible saltarse este paso instalando la distribución normal conteniendo `bootpd`. El demonio se puede ejecutar directamente utilizando el comando

```
bootpd -s
```

o a través de `inetd`. En ese caso será necesario:

1. editar `/etc/inetd.conf`, eliminando la marca de comentario (#) de las siguientes líneas:

   ```
   # tftp    dgram  udp  wait  root  /usr/sbin/in.tftpd   tftpd   /ex
   # bootps  dgram  udp  wait  root  /usr/sbin/in.bootpd  bootpd
   ```

2. inserte o descomente las siguientes dos líneas en el archivo `/etc/services`:

   ```
           bootps          67/tcp          # BOOTP server
           tftp            69/udp          # TFTP server
   ```

3. reiniciar `inetd` usando

   ```
   kill -HUP PID_de_inetd
   ```

6.2.1 Configuración de `bootpd` en el servidor

Lo primero de todo: `bootpd` tiene un archivo de configuración llamado `bootptab` el cual usualmente se encuentra en `/etc`. Debe ser modificado insertando la dirección IP de su pasarela, servidor de nombres y la/s dirección/es ethernet de su/s máquina/s sin disco. Un ejemplo de `/etc/bootptab`:

```
global.prof:\
    :sm=255.255.255.0:\
    :ds=192.168.1.5:\
    :gw=192.168.1.19:\
    :ht=ethernet:\
    :bf=linux:
machine1:hd=/export/root/machine1:tc=global.prof:ha=0000c0863d7a:ip=192.1
machine2:hd=/export/root/machine2:tc=global.prof:ha=0800110244e1:ip=192.1
machine3:hd=/export/root/machine3:tc=global.prof:ha=0800110244de:ip=192.1
```

- `global.prof`: Archivo de esquema general para las configuraciones de cada máquina, donde:

- `sm`: Corresponde a la Máscara de subred
- `ds`: Dirección del servidor de nombres (DNS)
- `gw`: Dirección de la pasarela por defecto
- `ht`: Tipo de hardware de red
- `bf`: Nombre del archivo de arranque

Después de esto, cada máquina debe tener una línea:

- El primer campo contiene el nombre de la máquina
- `hd` contiene el directorio del archivo de arranque
- el archivo de esquema general debe ser incluído en el campo `tc`
- `ha` contiene la dirección de hardware de la tarjeta de red
- `ip` contiene la dirección ip que se asigna

6.3 `tftp`

TFTP (*Trivial File Transfer Protocol*) es un protocolo de transferencia de archivos, como lo es ftp, pero mucho más simple, a fin de que quepa en las EPROMs. TFTP puede ser usado de dos maneras:

tftp simple:

El cliente puede acceder a la totalidad de su sistema de archivos. Es simple pero puede generar un gran problema de seguridad (cualquiera puede obtener su archivo de claves vía tftp).

tftp seguro:

El servidor `tftp` utiliza un llamada de sistema `chroot.2` para cambiar a su propio directorio raíz. Cualquier cosa que esté fuera del nuevo directorio raíz es completamente inaccesible, ya que el directorio que sufre el chroot se transforma en el nuevo directorio raíz; el campo `hd` del archivo `bootptab` debe reflejar esta nueva situación.

Por ejemplo: cuando se utiliza `tftp` simple, el campo `hd` contiene el *path* o camino completo del directorio de arranque: `/export/root/machine1`. Cuando se utiliza `tftp` seguro con `/export` como directorio raíz, entonces `/export` pasa a ser `/` y el campo `hd` debe ser `/root/machine1`.

Asimismo, cada implementación Unix contiene el servidor `tftp` y, probablemente, no sea necesario que Usted instale su propio servidor.

6.4 Servidor. Instalación mínima de Linux remota.

El servidor debe contener los paquetes *a*, *ap*, *n* y *x* de la distribución Slackware. No hay inconvenientes para instalar más cosas; sin embargo, los paquetes mencionados son suficientes para los propósitos de una terminal X sin disco.

Para la instalación se necesita un sistema Linux funcionando.

Buscar algo de espacio en disco para exportarlo para lectura-escritura en la máquina remota.

Montar el directorio exportado en algún lugar (por ej. `/mnt`) sobre el sistema de archivos de la máquina Linux.

Iniciar la instalación de Linux y cambiar el lugar de instalación de `/` a `/mnt`. Instalar los paquetes referidos de la forma usual. Si se desea correr sólo un Linux sin disco no son necesarios más cambios. De otra forma, si se planea utilizar más de una máquina sin disco esta instalación no funcionará ya que algunos archivos y directorios deben ser privados para cada máquina. El problema puede ser solucionado moviendo `/usr` (que no contiene elementos privados) y creando subdirectorios separados para cada máquina sin disco. Por ejemplo, si `/export/linux/machine1` fue montado en `/mnt` entonces la estructura de directorios después de la instalación inicial se verá así:

```
/export/linux/machine1/bin
/export/linux/machine1/sbin
/export/linux/machine1/lib
/export/linux/machine1/etc
/export/linux/machine1/var
/export/linux/machine1/usr
```

Y después de los cambios debe de quedar:

```
/export/linux/machine1/bin
/export/linux/machine1/sbin
/export/linux/machine1/lib
/export/linux/machine1/etc
/export/linux/machine1/var
/export/linux/usr
```

Ahora es posible crear los subdirectorios para las otras máquinas. Asumimos que las máquinas sin disco son llamadas `machine1`, `machine2`, `machine3`, etc.; puede utilizar el siguiente macro de bash para instalar los otros directorios:

```
cd /export/linux
for x in machine2 machine3 ; do
        mkdir $x; cd $x
        (cd ../machine1; tar cf - *) | tar xvf -
done
```

Una vez hecho esto, realice las siguientes exportaciones:

- `/export/linux/usr`: Sólo lectura para todo el mundo
- `/export/linux/machine1`: Solamente para `machine1` con permisos `rw,root`
- `/export/linux/machine2`: Solamente para `machine2` con derechos `rw,root`
- `/export/linux/machine3`: Solamente para `machine3` con derechos `rw,root`

como se indica a continuación (el formato de este ejemplo está en la sintaxis de SunOs 4.1.3 para `export`)

```
# Este archivo es /etc/export
# para terminales X con Linux remotas
# La siguiente linea es una unica:
/etc/root/usr                   -access=linuxnet
# Las siguientes lineas corresponden a una por cada maquina
/export/root/machine1           rw=machine1,root=machine1
/export/root/machine2           rw=machine2,root=machine2
/export/root/machine3           rw=machine3,root=machine3
```

No olvidar ejecutar `exportfs -a`

6.5 Configuración del servidor `tftp`

Ahora es el momento de configurar el servidor `tftp`. Si no necesita seguridad en el protocolo `tftp` todo resultará más simple para sus clientes ya que arrancarán desde el directorio `/export`.

Si utiliza seguridad con el protocolo `tftp` entonces se debe crear una estructura de directorio completa `/export/linux` bajo `/tftpboot` (con un núcleo real y enlaces simbólicos para las otras máquinas), o dejar que el directorio `/export` sea el directorio de arranque para el `tftp` seguro. O, si dispone de un directorio `tftpboot` separado necesita, de igual forma, la estructura de directorios original con un núcleo único y enlaces simbólicos para los otros. Se puede conseguir esta configuración ejecutando lo siguiente:

```
mkdir -p /tftpboot/export/linux/machine1
cd /tftpboot/export/linux/machine1
cp /export/linux/machine1/nombre_del_nucleo
```

Teclee lo siguiente:

```
mkdir -p /tftpboot/export/linux/machine2
cd ../machine2
ln -s ../machine1/nombre_del_nucleo
```

6.6 Últimos retoques

Para terminar, se debe insertar:

```
/sbin/mount nombre_servidor_nfs:/export/linux/usr /usr
```

como la primera línea de

```
/export/linux/machineX/etc/rc.d/rc.S
```

6.7 Requerimientos de memoria y espacio en disco; velocidad

Todo lo indicado ha sido comprobado únicamente con Slackware 2.3; para otras distribuciones/versiones los datos que se indican a continuación pueden variar.

- Espacio en disco: 28Mb + 6.5Mb/máquina
- RAM: Yo utilizo X con 8Mb. Para lo cual se necesita 4Mb de swap, el cual puede ser creado, separadamente para cada máquina, mediante un fichero de swap en el directorio `/tmp`. No olvide ponerlo a punto con `mkswap` y `swapon`.
- Velocidad: No he tenido problemas con un 486 DX2/66 con 8 Mb.

7 Posibles errores

- Me encontré con un extraño error: SunOS corrompió los dispositivos del directorio `/dev`, por lo que me fue necesario ejecutar `MAKEDEV` montando los subdirectorios en un disco con Linux (La razón fueron las diferencias entre el nfs de Linux y el de SunOS: ambos utilizan 32 bit para el número de dispositivo Mayor y Menor, pero mientras Linux usa campos de tamaño 16 bit para ambos, SunOS utiliza 14 bit para el Mayor y 18 bit para el número Menor de dispositivo)

- Cuando la máquina Linux sin disco comienza el arranque, existe solamente una ruta incluida en la tabla de rutas del servidor `tftp`, por lo que es necesario configurar una tabla correcta de rutado; se disponen de dos opciones:

  1. configurar el fichero `rc.S` para cada máquina a mano
  2. usar un cliente `bootp` y escribir un macro de configuración

8 Errores y posibles ampliaciones futuras de este documento

- Corrige citas de los documentos referidos
- SunOS está basado en BSD. Necesita la inclusión de una configuración basada en SVR4 (por ej. Solaris).
- Ya que Linux es muy semejante a SunOS en sus servidores `bootp` y `tftp`, el ejemplo de servidor basado en Linux podría ser de utilidad.
- Poner al día este documento con el paquete actual `etherboot`.
- Mostrar las diferencias entre núcleos con nfs-root incluido como parche y núcleos, posteriores al 1.3.x, que ya contengan nfs-root como opción.

9 Anexo: El INSFLUG

El *INSFLUG* forma parte del grupo internacional *Linux Documentation Project*, encargándose de las traducciones al castellano de los Howtos (Comos), así como la producción de documentos originales en aquellos casos en los que no existe análogo en inglés.

En el **INSFLUG** se orienta preferentemente a la traducción de documentos breves, como los *COMOs* y *PUFs* (**P**reguntas de **U**so **F**recuente, las *FAQs*. `:)`), etc.

Diríjase a la sede del INSFLUG para más información al respecto.

En la sede del INSFLUG encontrará siempre las **últimas** versiones de las traducciones: `www.insflug.org`. Asegúrese de comprobar cuál es la última versión disponible en el Insflug antes de bajar un documento de un servidor réplica.

Se proporciona también una lista de los servidores réplica (*mirror*) del Insflug más cercanos a Vd., e información relativa a otros recursos en castellano.

Francisco José Montilla, `pacopepe@insflug.org`.