SLUG talk: Setting up a diskless Linux system

Ken Yap

1. Why network booting?

- No moving parts, less noise and power consumption
- Centralised administration
- No tampering
- Standalone applications

2. How does it work?

1. Diskless computer (DC) broadcasts MAC address with bootp: Who am I?
2. DHCP server on S looks up DB: Your IP address is X.X.X.X, your server is S, your boot file is `vmlinuz.myname`, etc.
3. DC asks to load file from TFTP server on S: Please give me `vmlinuz.myname`
4. S: Here you are (`/tftpdir/vmlinuz.myname`)

   DC thinks a while (booting Linux).

5. DC: Please let me mount `/` with NFS
6. DC: Please let me mount other NFSes (`/usr, /home/`, etc)
7. S: Here you are
8. DC: Runs intended application

Network boot ROM contains code to do 1 and 3.

3. Server setup

3.1. Bootpd setup

- (In 2003: DHCPD is preferred.) Install bootpd
- Edit `/etc/bootptab`, typical line

```
xterm.ken.net.au:tc=.default:ht=ethernet:ha=08002BB7F380:\
 ip=192.168.0.100:bf=vmlinuz.xterm
```

3.2. Tftpd setup

- Install tftpd, make sure it's active in `/etc/inetd.conf`, typical line

```
tftp dgram udp wait root /usr/sbin/tcpd in.tftpd /tftpdir
```

3.3. Kernel image

(In 2003: The preferred method is to download along with the kernel a ramdisk that sets up a initial root filesystem, which is then used to mount the real root filesystem. www.ltsp.org does it this way.)

After building the kernel and ramdisk, run `mkelf-linux` from the Etherboot distribution on it. Install this *boot image* as `/tftpdir/<bf attribute in bootptab>`.

4. Net loader

A small program that runs as a BIOS extension, usually on an EPROM on the NIC. It handles the BOOTP query and TFTP loading and then transfers control to the loaded image.

It uses UDP/IP protocols but the loaded image doesn't have to be Linux. The loaded image can be anything, even DOG.

It can also be loaded from a floppy for testing and for temporary setups.

5. RH5

(In 2003: Details of RH5 setup too old to recommend and therefore removed.)

6. X-terminal

The reason I started this. I had a 486DX2/66 left over from an upgrade and I wanted to have a quiet X-terminal while the server makes noise somewhere else.

486 or Pentium best. 16 MB should be the minimum. Put in the best video card you can get for it. Normal NIC should do.

On the server, make sure the DC is matched by a clause in `/etc/X11/xdm/Xaccess` and comment out the `:0` in `/etc/X11/xdm/Xservers`. Then make sure that xdm is run from the init scripts.

On the client, run `X -query` *server*

You will get the xdm login box and then all your X clients will run on the server.

I have been using this for several weeks now.

7. Other applications

You could netboot routers, print servers (but should not be spooling print server), standalone apps, etc.