

Network Bootloading with PXE

Andrew Beard

6 June, 2002

Introduction

Preboot eXecution Environment is a standard developed by Intel to boot and install operating systems over a network. The basic requirement is a PXE compliant network card integrated into the standard pc bios (most equipment manufactured after 1999 fit this requirement). In addition, a typical PXE network/software environment will require a DHCP server, TFTP server and a 2nd stage bootloader with which to load the chosen operating system kernel. In the following description we use ISC DHCP 2.0, TFTP, and PXELINUX version 1.73 (included with the SYSLINUX package).

The basic sequence of events when booting via PXE is as follows...

- a) A PXE enabled PC is turned on and initiates a DHCP request to the server containing the MAC address of the client nic card.
- b) The server responds with the clients IP address, and the location of the tftp server as well as the location of the 2nd stage bootloader executable on that server.
- c) The client proceeds to download the 2nd stage bootloader image using tftp and loads the binary into memory.
- d) At this stage PXE is essentially done and the 2nd stage bootloader takes over using a universal generic network card driver.
- e) The second stage bootloader then downloads a kernel image and proceeds to execute it while passing any configuration specific parameters to the kernel.
- f) The kernel takes over and attempts to finish booting the machine... At this stage the kernel is in complete control and proceeds to boot in the normal manner.

Detailed Installation

- 1) Configure client pc bios.
- 2) Configure the server and 2nd stage bootloader.
- 3) Build a properly configured Linux kernel with nfs root.

Configuring client bios

To enable the client pc PXE must be enabled in the bios. This can usually be done by setting a network boot option in the standard boot sequence (standard cmos options) or by enabling PXE in the integrated devices section of the bios. The PXE option may be hidden under a larger class of network boot methods or MBA (managing boot agents) so it may be necessary to read the bios documentation for the particular machine. This should be all that is necessary to configure the client.

Server configuration

Server side configuration requires several steps...

DHCP server configuration

First the DHCP server must be configured with the following options in the dhcpd.conf file ...

```

group {
    filename "pxelinux.0"
    option dhcp-class-identifier "PXEClient";
    option vendor-encapsulated-options 01:04:00:00:0:00;

    host bodie {
        next server sabrina;
        hardware ethernet 00:06:5b:1c:f1:3a;
        fixed address 192.100.16.83;
    }

    host xxxx
}

```

In this configuration, filename specifies the name of the second stage bootloader on the tftp server (specified in the host section). This file should reside in the default tftp folder on the tftp server (specified in the xinetd tftp configuration file on sabrina). It is downloaded to the client machine after the initial pxe/dhcp requests are made and proceeds with the boot process. Bodie is the name of our sample client machine. The next server option contains the hostname of the tftp server. The hardware ethernet field contains the mac address of the nic card within the client – this is usually obtained from the pxe configuration menu within the client bios options. The fixed address is the ip address of the client machine as listed within the /etc/hosts file. Note that for this particular setup, the TFTP server, NFS file server and DHCP server are all the same – sabrina.

Configuring PXELINUX

Several files need to be edited to configure the PXELINUX 2nd stage bootloader. PXELINUX is very similar to LILO and once downloaded to the client machine performs essentially the same function. The configuration files are client specific and contained in the default tftpboot folder within a subfolder titled pxelinux.cfg. PXELINUX expects a configuration file named after the hex ip address of the client. When started on the client PXELINUX searches for the full hex IP address file first, if it isn't found it will begin looking for files named with a subset of this IP address with trailing digits sequentially dropped. Using this method it is possible to have a single configuration file for multiple clients which share a base IP address. Finally if none of the aforementioned files are found, PXELINUX searches for a file called Default. Thus if the hex IP address of a client machine is C0641053, PXELINUX will search for files named C0641053, C064105, C06410...C and finally Default in that order until it is found. The format of the configuration files contains several things, namely a LABEL for a particular kernel image, the kernel name (i.e. bzImage) and any options that need to be appended to the kernel command line. The configuration file for bodie contains the following...

LABEL bodie

 KERNEL bzBodieImage

 APPEND nfsaddr=192.100.16.29 root= /tftpboot/192.100.16.83

The kernel image should reside in the default tftpboot directory. The nfsaddr append statement tells the kernel to mount its' root filesystem using the specified nfs server ip address and the root option tells it where on this server the root filesystem is located.

Configuring a nfs mounted root filesystem

If PXE is to be used for diskless netbooting, a nfs mounted root filesystem must be enabled and nfs root capabilities must be built into the kernel. This can be done in several ways to ensure that common client files and folders are shared and client specific folders are unique. For more information on this the reader is referred to the NFS-Root-Client Mini-HOWTO (v4.1, 02 Feb, 1999). Once the nfs root filesystem is setup and exported network booting should be ready to proceed.

Conclusion

This should be it, at this stage simply start the client machine and let the boot process take place.