

- Partners
- Support
- Community
- Ubuntu.com

- Page History
- Login to edit

Search

PXEInstallMultiDistro

This tutorial shows how to set up a PXE (short for pre-boot execution environment) install server with Ubuntu 9.10 (Karmic Koala).

A PXE install server allows your client computers to boot and install a Linux distribution over the network, without the need of burning Linux ISO images onto a CD/DVD, boot floppy images, etc. This is handy if your client computers don't have CD or floppy drives, or if you want to set up multiple computers at the same time (e.g. in a large enterprise), or simply because you want to save the money for the CDs/DVDs.

In this article I show how to configure a PXE server that allows you to boot multiple distributions: Ubuntu, Debian, Fedora, CentOS, and openSUSE.

The end result will be a network bootable menu with sub-menus allowing you to choose an OS to install/boot:

Contents

- 1. Getting Started
- 2. Install prerequisite software
 - 1. TFTP Server
 - 2. DHCP Server
 - 3. SYSLINUX
 - 4. NFS
- 3. Putting it All Together
 - 1. Fedora
 - 2. openSUSE
 - 3. CentOS
 - 4. Ubuntu
 - 5. DOS
 - 6. Utilities
- 4. Summary
- 5. Troubleshooting





Required

- TFTP Server
- Syslinux
- DHCP Server
- NFS Server
- PXE capable NICs

Optional

- Apache
- Apt-mirror
- BIND

If you intend on installing Ubuntu via PXE/HTTP, I highly recommend apt-mirror is installed to provide a local repository of Ubuntu packages and Apache is installed to serve those packages via HTTP.

BIND will allow the DHCP server to assign host names to the PXE clients. This article will not cover the installation and configuration of BIND, however to assign dynamic names to DHCP clients I have the following in my zone files:

Forward look-up zone file entry:
\$GENERATE 100-200 dhcp-\$ IN A 10.10.1.\$

Reverse look-up zone file entry:
\$GENERATE 100-200 \$ IN PTR dhcp-\$.home.local.

Getting Started

Install Ubuntu 9.10 (Karmic Koala) on a system that you will use as the deployment server. I prefer a minimal install and then install required/optional packages as needed. Once your operating system is installed, update the system with the package manager of your choice.

```
sudo apt-get update
sudo apt-get dist-upgrade
sudo apt-get clean
```

These can be included in a single command: `sudo apt-get update && sudo apt-get dist-upgrade && sudo apt-get clean` (&& will run the next command only if the previous command succeeds)

Install prerequisite software

TFTP Server

```
sudo apt-get -y install tftpd-hpa
sudo /etc/init.d/openbsd-inetd stop
sudo update-rc.d -f openbsd-inetd remove
sudo sed -i s/no/yes/ /etc/default/tftpd-hpa
sudo /etc/init.d/tftpd-hpa start
```

Ubuntu installs the openbsd-inetd package when the tftpd-hpa package is installed. In our example we will simply run TFTP as a daemon and will always be listening for connections.

In the above code snippet:

- tftpd-hpa was installed

- The openbsd-inetd daemon was stopped
- openbsd-inetd was removed from the startup scripts
- /etc/default/tftpd-hpa was modified to allow tftpd-hpa to run as a daemon process
- tftpd-hpa was started

Verify the TFTP server is listening for connections:

```
ss -apu | grep tftp
```

Sample output:

```
UNCONN 0 0 *:tftp *:*
```

DHCP Server

If your network does not have a DHCP server, install the Internet Software Consortium's DHCP server. With an existing DHCP server, you will need to modify the scope for your subnet to include the information allowing PXE clients to contact the TFTP server.

The scope of an existing DHCP server should be modified to reflect:

```
filename "pxelinux.0";
next-server 10.10.1.10;
```

I will assume no DHCP server exists. We will install and configure one.

```
sudo apt-get -y install dhcp3-server
```

After installation the daemon will attempt to start, however, will fail since the default configuration file does not contain a valid DHCP scope for our subnet. The configuration file for dhcp3-server is /etc/dhcp3/dhcpd.conf. The file is well commented and easily modified to fit your subnet(s).

Example /etc/dhcp3/dhcpd.conf:

```
ddns-update-style none;
option domain-name "home.local";
option domain-name-servers 10.10.1.10;
default-lease-time 86400;
max-lease-time 604800;
option time-offset -18000;
authoritative;
log-facility local7;
allow booting;
allow bootp;
subnet 10.10.1.0 netmask 255.255.255.0 {
    get-lease-hostnames on;
    use-host-decl-names on;
    range 10.10.1.100 10.10.1.200;
    option routers 10.10.1.1;
    option subnet-mask 255.255.255.0;
    option broadcast-address 10.10.1.255;
    filename "pxelinux.0";
    next-server 10.10.1.10;
}
host shuttle {
    hardware ethernet 00:30:1b:ba:89:31;
    fixed-address 10.10.1.20;
    option routers 10.10.1.1;
    option subnet-mask 255.255.255.0;
    option broadcast-address 10.10.1.255;
    option host-name "shuttle";
    filename "pxelinux.0";
    option root-path "/home/shuttle";
    next-server 10.10.1.10;
}
host nfsroot {
    hardware ethernet 08:00:27:bb:74:dd;
    fixed-address 10.10.1.30;
    option routers 10.10.1.1;
    option subnet-mask 255.255.255.0;
    option broadcast-address 10.10.1.255;
    option host-name "nfsroot";
    filename "pxelinux.0";
    option root-path "/home/nfsroot";
    next-server 10.10.1.10;
}
```

In the above configuration:

1. Dynamic DNS Updates are disabled
2. The DNS domain name is home.local
3. The DNS server is 10.10.1.10 (this is also the TFTP and NFS server)
4. The DHCP lease time is 1 day
5. The log will include a time-offset of -18000 to more closely match the local time zone. The DHCP client leases can be viewed in /var/lib/dhcp3/dhcpd.leases
6. The DHCP server is authoritative for the network
7. Booting is enabled, this does not imply PXE booting, however other documents will be written to support booting a file-system over NFS instead of a local disk.
8. The DHCP scope of the subnet for 10.10.1.1 includes:
 1. A range of addresses to be assigned to DHCP client
 2. The location of the TFTP server
 3. The file to obtain from the DHCP server
9. Various hosts are reserved IP addresses in order for them to always receive the same IP address via DHCP.

Start the DHCP server.

```
sudo /etc/init.d/dhcp3-server start
```

Verify the DHCP server is listening for client connections.

```
ss -apu | grep dhcpd
```

Sample output:

```
UNCONN 0 0 *:bootps *:users:("dhcpd3",...)
```

SYSLINUX

The SYSLINUX Project is a suite of lightweight boot-loaders, for starting up computers with the Linux kernel. It is the work of H. Peter Anvin, and consists of several separate systems, the best-known of which is ISOLINUX.

The PXELINUX bootstrap will be installed when syslinux is installed.

PXELINUX is used in conjunction with a PXE compliant ROM on a network card. The PXE environment uses DHCP or BOOTP to enable basic TCP/IP networking, then downloads a bootstrap program via TFTP. This bootstrap program loads and configures a kernel according to directives that are also downloaded from the TFTP server.

Typically, PXELINUX is used for Linux installations from a central network server or for booting disk-less workstations.

Install SYSLINUX.

```
sudo apt-get -y install syslinux
```

Copy the PXELINUX bootstrap to the root of our TFTP server.

```
sudo cp /usr/lib/syslinux/pxelinux.0 /var/lib/tftpboot
```

Configuration files for PXELINUX reside in directory /var/lib/tftpboot/pxelinux.cfg/. PXELINUX uses the following method to search for the appropriate configuration file:

- The hardware type (using its ARP type code) and address, all in lower case hexadecimal with dash separators; for example, for an Ethernet (ARP type 1) with address 88:99:AA:BB:CC:DD it would search for the file-name 01-88-99-aa-bb-cc-dd.
- The client IP address in upper case hexadecimal, e.g. 192.0.2.91 -> C000025B
- Continuously remove one hex digit from the hexadecimal IP address
- A file named default

As an example, if the boot file name is pxelinux.0, the Ethernet MAC address is 88:99:AA:BB:CC:DD and the IP address 192.0.2.91, it will try following the files:

- /var/lib/tftpboot/pxelinux.cfg/01-88-99-aa-bb-cc-dd
- /var/lib/tftpboot/pxelinux.cfg/C000025B
- /var/lib/tftpboot/pxelinux.cfg/C000025
- /var/lib/tftpboot/pxelinux.cfg/C00002
- /var/lib/tftpboot/pxelinux.cfg/C0000
- /var/lib/tftpboot/pxelinux.cfg/C000
- /var/lib/tftpboot/pxelinux.cfg/C00
- /var/lib/tftpboot/pxelinux.cfg/C0
- /var/lib/tftpboot/pxelinux.cfg/C
- /var/lib/tftpboot/pxelinux.cfg/default

Create the PXELINUX default configuration file.

```
sudo touch /var/lib/tftpboot/pxelinux.cfg/default
```

NFS

There are two NFS servers for Ubuntu.

1. nfs-user-server
2. nfs-kernel-server

As the package name indicates, one runs in user space and the other in kernel space.

Install NFS.

```
sudo apt-get -y install nfs-kernel-server
```

The NFS server uses /etc/exports to identify what local directories are available to NFS clients.

We will be using /srv/install and the NFS export to store operating system files used for installation.

```
sudo mkdir /srv/install
```

Modify /etc/exports and make /srv/install available for our Linux installations.

Example /etc/exports:

```
/srv/install 10.10.1.0/24(ro,async,no_root_squash,no_subtree_check)
```

The above configuration will allow read-only access via NFS to /srv/install/ for clients on the 10.10.1.0 network.

Export our file system or restart the NFS server.

```
sudo exportfs -a
or
sudo /etc/init.d/nfs-kernel-server restart
```

Putting it All Together

We have installed:

1. TFTP Server
2. SYSLINUX
3. NFS Server
4. DHCP Server

Let's start our first install with Fedora 10 64-bit. You will need to make the kernel and initrd available via TFTP, and the installation media available via NFS.

Choose a directory structure that will allow you to maintain multiple distributions.

As an example:

To store the kernel and initrd:

```
/var/lib/tftpboot/fedora/12/i386
/var/lib/tftpboot/fedora/12/amd64
/var/lib/tftpboot/CentOS/5.4/amd64
/var/lib/tftpboot/CentOS/5.4/i386
/var/lib/tftpboot/Ubuntu/9.10/i386
/var/lib/tftpboot/Ubuntu/9.10/amd64
/var/lib/tftpboot/openSUSE/11.2/i386
/var/lib/tftpboot/openSUSE/11.2/amd64
```

To store the installation media:

```
/srv/install/fedora/12/i386
/srv/install/fedora/12/amd64
/srv/install/CentOS/5.4/amd64
/srv/install/CentOS/5.4/i386
/srv/install/Ubuntu/9.10/i386
/srv/install/Ubuntu/9.10/amd64
/srv/install/openSUSE/11.2/i386
/srv/install/openSUSE/11.2/amd64
```

Fedora

Create the directories to store Fedora 10.

```
sudo mkdir -p /var/lib/tftpboot/fedora/12/i386
sudo mkdir -p /var/lib/tftpboot/fedora/12/amd64
sudo mkdir -p /srv/install/fedora/12/i386
sudo mkdir -p /srv/install/fedora/12/amd64
```

Mount the Fedora 12 64-bit DVD ISO and copy the kernel and initrd to the previously created location.

```
sudo mkdir /mnt/loop
sudo mount -o loop -t iso9660 /location/of/ISO/Fedora-12-x86_64-DVD.iso /mnt/loop
sudo cp /mnt/loop/images/pxeboot/vmlinuz /var/lib/tftpboot/fedora/12/amd64
sudo cp /mnt/loop/images/pxeboot/initrd.img /var/lib/tftpboot/fedora/12/amd64
sudo cp -R /mnt/loop/* /srv/install/fedora/12/amd64
sudo umount /mnt/loop
```

Mount the Fedora 12 32-bit DVD ISO and copy the kernel and initrd to the previously created location.

```
sudo mkdir /mnt/loop
sudo mount -o loop -t iso9660 /location/of/ISO/Fedora-12-i386-DVD.iso /mnt/loop
sudo cp /mnt/loop/images/pxeboot/vmlinuz /var/lib/tftpboot/fedora/12/i386
sudo cp /mnt/loop/images/pxeboot/initrd.img /var/lib/tftpboot/fedora/12/i386
sudo cp -R /mnt/loop/* /srv/install/fedora/12/i386
sudo umount /mnt/loop
```

Many options exist for PXELINUX. You can have:

1. No menu
2. An abysmal black screen with text
3. A menu
4. An advanced menu that supports nested menus
5. A graphical menu

Since we want a nice pretty menu that contains sub-menus and a background image, we will need vesamenu.c32 from the SYSLINUX installation and our background image. (logo.png)

```
sudo cp /usr/lib/syslinux/vesamenu.c32 /var/lib/tftpboot/
sudo cp /location/of/image/logo.png /var/lib/tftpboot/pxelinux.cfg/
```

Modify our PXELINUX configuration file.

Example /var/lib/tftpboot/pxelinux.cfg/default:

```
DEFAULT vesamenu.c32
TIMEOUT 600
ONTIMEOUT BootLocal
PROMPT 0
MENU INCLUDE pxelinux.cfg/pxe.conf
NOESCAPE 1
LABEL BootLocal
    localboot 0
    TEXT HELP
    Boot to local hard disk
    ENDTEXT
MENU BEGIN Ubuntu
MENU TITLE Ubuntu
    LABEL Previous
    MENU LABEL Previous Menu
    TEXT HELP
    Return to previous menu
    ENDTEXT
    MENU EXIT
    MENU SEPARATOR
    MENU INCLUDE Ubuntu/Ubuntu.menu
MENU END
MENU BEGIN Redhat Enterprise Linux
MENU TITLE Redhat Enterprise Linux
    LABEL Previous
    MENU LABEL Previous Menu
    TEXT HELP
    Return to previous menu
    ENDTEXT
    MENU EXIT
    MENU SEPARATOR
    MENU INCLUDE RHEL/RHEL.menu
MENU END
MENU BEGIN CentOS
MENU TITLE CentOS
    LABEL Previous
    MENU LABEL Previous Menu
    TEXT HELP
    Return to previous menu
    ENDTEXT
    MENU EXIT
    MENU SEPARATOR
    MENU INCLUDE CentOS/CentOS.menu
MENU END
MENU BEGIN Fedora
MENU TITLE Fedora
    LABEL Previous
    MENU LABEL Previous Menu
    TEXT HELP
    Return to previous menu
    ENDTEXT
    MENU EXIT
    MENU SEPARATOR
    MENU INCLUDE Fedora/Fedora.menu
MENU END
MENU BEGIN openSUSE
MENU TITLE openSUSE
    LABEL Previous
    MENU LABEL Previous Menu
    TEXT HELP
    Return to previous menu
    ENDTEXT
    MENU EXIT
    MENU SEPARATOR
    MENU INCLUDE openSUSE/openSUSE.menu
MENU END
MENU BEGIN openFile Storage Server
MENU TITLE openFile Storage Server
    LABEL Previous
    MENU LABEL Previous Menu
    TEXT HELP
    Return to previous menu
    ENDTEXT
    MENU EXIT
    MENU SEPARATOR
    MENU INCLUDE Openfiler/Openfiler.menu
MENU END
MENU BEGIN VMware
MENU TITLE VMware
    LABEL Previous
    MENU LABEL Previous Menu
    TEXT HELP
    Return to previous menu
    ENDTEXT
    MENU EXIT
    MENU SEPARATOR
    MENU INCLUDE VMware/VMware.menu
MENU END
MENU BEGIN Tools and Utilities
MENU TITLE Tools and Utilities
```

```

        LABEL Previous
        MENU LABEL Previous Menu
        TEXT HELP
        Return to previous menu
        ENDTEXT
        MENU EXIT
        MENU SEPARATOR
        MENU INCLUDE utilities/utilities.menu
MENU END
MENU BEGIN DOS Based
MENU TITLE DOS Based
        LABEL Previous
        MENU LABEL Previous Menu
        TEXT HELP
        Return to previous menu
        ENDTEXT
        MENU EXIT
        MENU SEPARATOR
        MENU INCLUDE dos/dos.menu
MENU END

```

An explantion of the example `/var/lib/tftpboot/pxelinux.cfg/default`:

- **DEFAULT vesamenu.c32** - We are informing PXELINUX to load this file. We could substitute a menu entry instead, or boot a kernel and initrd.
- **TIMEOUT 600** - 600 seconds will pass before action is taken on the users behalf if no keys are pressed
- **ONTIMEOUT** - Indicates what action is performed when the TIMEOUT expires. In the above example, we are loading the menu item BootLocal. If after 5 minutes the user does not choose a menu item, the system will attempt to boot to a local hard disk.
- **PROMPT 0** - Will allow PXELINUX to prompt us for input
- **MENU INCLUDE pxelinux.cfg/pxe.conf** - Load additional configuration options from another file. The can be included in the same file, however, I find it a neater look to the menu configuration to place some options in other files.
- **NOESCAPE 1** - Do not allow the user to escape out of the menu system.
- **MENU BEGIN** - Start a new menu
- **MENU TITLE** - Display title for menu
- **LABEL** - Refer to this menu by label
- **MENU LABEL** - Display title for label
- **TEXT HELP** - Optional text displayed on screen when highlighted
- **ENDTEXT** - End of optional text
- **MENU SEPARATOR** - Inserts a blank non-selectable line
- **MENU INCLUDE** - Include items from another file and create a menu from them. This is used to create the sub-menus
- **MENU END** - End of a menu item

Example `pxelinux.cfg/pxe.conf`:

```

MENU TITLE PXE Server
MENU BACKGROUND pxelinux.cfg/logo.png
NOESCAPE 1
ALLOWOPTIONS 1
PROMPT 0
menu width 80
menu rows 14
MENU TABMSGROW 24
MENU MARGIN 10
menu color border          30;44      #ffffff #00000000 std

```

The above `pxe.conf` file is configuring

1. The colors used
2. The width
3. The number of items displayed
4. The backgroud image

As we are using Fedora 10 as our first example operating system, the following is the Fedora menu.

Create the Fedora menu.

```
touch /var/lib/tftpboot/fedora/fedora.menu
```

Example `/var/lib/tftpboot/fedora/fedora.menu`:

```

LABEL 2
    MENU LABEL Fedora 12 (64-bit)
    KERNEL fedora/12/amd64/vmlinuz
    APPEND method=nfs:10.10.1.10:/srv/install/fedora/12/amd64/ lang=us keymap=us ip=dhcp ksdevice=eth0 noip6 initrd=fedora/12/i
    TEXT HELP
    Install Fedora 12 (64-bit)
    ENDTEXT
LABEL 1
    MENU LABEL fedora 12 (32-bit)
    KERNEL fedora/12/i386/vmlinuz
    APPEND method=nfs:10.10.1.10:/srv/install/fedora/12/i386/ lang=us keymap=us ip=dhcp ksdevice=eth0 noip6 initrd=fedora/12/i
    TEXT HELP
    Install Fedora 12 (32-bit)
    ENDTEXT

```

The above menu entry for Fedora 12 informs the client:

1. Which kernel to load
2. The location of the installation media

3. The language used during the installation
4. The keyboard map used during the installation
5. Obtain an IP address using DHCP
6. Use eth0 to install the operating system in case there are multiple NICs
7. Disable IPv6
8. Which initrd to use
9. The ram disk size used during the install

This does not prevent every installation question from being asked. You will still be prompted for:

1. Disk partitioning scheme
2. Package selection
3. User account information
4. Etc.

You can completely automate the installation using a kickstart file, however, this article will not cover its usage.

When a client now boots via PXE, the client will:

1. Request an IP address
2. The server will assign an IP address and inform the client which bootloader to use i.e pxelinux.0.
3. The Client will download pxelinux.0 via TFTP
4. pxelinux.0 will load vesamenu.c32
5. The graphical menu will be displayed on the screen.
6. Once the user navigates to the Fedora menu and chooses the version to install:
 1. The kernel and initrd will be downloaded via TFTP
 2. Control will be handed over to the kernel
 3. Mount the NFS export and the installation process will begin.

openSUSE

Next, we will follow the same procedure and configure support for openSUSE to be installed.

Create the directories to store openSUSE 11.2.

```
sudo mkdir -p /var/lib/tftpboot/openSUSE/11.2/i386
sudo mkdir -p /var/lib/tftpboot/openSUSE/11.2/amd64
sudo mkdir -p /srv/install/openSUSE/11.2/i386
sudo mkdir -p /srv/install/openSUSE/11.2/amd64
```

Mount the openSUSE 11.2 64-bit DVD ISO and copy the kernel and initrd to the previously created location.

```
sudo mkdir /mnt/loop
sudo mount -o loop -t iso9660 /location/of/ISO/openSUSE-11.2-DVD-x86_64.iso /mnt/loop
sudo cp /mnt/loop/boot/x86_64/loader/linux /home/tftpboot/openSUSE/11.2/amd64
sudo cp /mnt/loop/boot/x86_64/loader/initrd /home/tftpboot/openSUSE/11.2/amd64
sudo cp -R /mnt/loop/* /srv/install/openSUSE/11.2/amd64
sudo umount /mnt/loop
```

Mount the openSUSE 11.2 32-bit DVD ISO and copy the kernel and initrd to the previously created location.

```
sudo mkdir /mnt/loop
sudo mount -o loop -t iso9660 /location/of/ISO/openSUSE-11.2-DVD-i586.iso /mnt/loop
sudo cp /mnt/loop/boot/i386/loader/linux /home/tftpboot/openSUSE/11.2/i386
sudo cp /mnt/loop/boot/i386/loader/initrd /home/tftpboot/openSUSE/11.2/i386
sudo cp -R /mnt/loop/* /srv/install/openSUSE/11.2/i386
sudo umount /mnt/loop
```

Create the openSUSE menu.

```
touch /var/lib/tftpboot/openSUSE/openSUSE.menu
```

Example /var/lib/tftpboot/openSUSE/openSUSE.menu

```
LABEL 2
MENU LABEL openSUSE 11.2 (64-bit)
KERNEL openSUSE/11.2/amd64/linux
APPEND initrd=openSUSE/11.2/amd64/initrd install=nfs://10.10.1.10/srv/install/openSUSE/11.2/amd64 splash=silent ramdisk_size=
TEXT HELP
Install openSUSE 11.2 (64-bit)
ENDTEXT

LABEL 1
MENU LABEL openSUSE 11.2 (32-bit)
KERNEL openSUSE/11.2/i386/linux
APPEND initrd=openSUSE/11.2/i386/initrd install=nfs://10.10.1.10/srv/install/openSUSE/11.2/i386 splash=silent ramdisk_size=
TEXT HELP
Install openSUSE 11.2 (32-bit)
ENDTEXT
```

Once again, this does not prevent every installation question from being asked. This article does not cover automated installations for openSUSE.

CentOS

Again we will follow the same procedure and configure support for CentOS to be installed.

Create the directories to store CentOS 5.4.


```
sudo mkdir -p /var/lib/tftpboot/CentOS/5.4/i386
sudo mkdir -p /var/lib/tftpboot/CentOS/5.4/amd64
sudo mkdir -p /srv/install/CentOS/5.4/i386
sudo mkdir -p /srv/install/CentOS/5.4/amd64
```

Mount the CentOS 5.4 64-bit DVD ISO and copy the kernel and initrd to the previously created location.

```
sudo mkdir /mnt/loop
sudo mount -o loop -t iso9660 /location/of/ISO/CentOS-5.4-x86_64-bin-DVD.iso /mnt/loop
sudo cp /mnt/loop/images/pxeboot/vmlinuz /var/lib/tftpboot/CentOS/5.4/amd64
sudo cp /mnt/loop/images/pxeboot/initrd.img /var/lib/tftpboot/CentOS/5.4/amd64
sudo cp -R /mnt/loop/* /srv/install/CentOS/5.4/amd64
sudo umount /mnt/loop
```

Mount the CentOS 5.4 32-bit DVD ISO and copy the kernel and initrd to the previously created location.

```
sudo mkdir /mnt/loop
sudo mount -o loop -t iso9660 /location/of/ISO/CentOS-5.4-i386-bin-DVD.iso /mnt/loop
sudo cp /mnt/loop/images/pxeboot/vmlinuz /var/lib/tftpboot/CentOS/5.4/i386
sudo cp /mnt/loop/images/pxeboot/initrd.img /var/lib/tftpboot/CentOS/5.4/i386
sudo cp -R /mnt/loop/* /srv/install/CentOS/5.4/i386
sudo umount /mnt/loop
```

Create the CentOS menu.

```
touch /var/lib/tftpboot/CentOS/CentOS.menu
```

Example /var/lib/tftpboot/CentOS/CentOS.menu:

```
LABEL 2
    MENU LABEL CentOS 5.4 (64-bit)
    KERNEL CentOS/5.4/amd64/vmlinuz
    APPEND method=nfs:10.10.1.10:/srv/install/CentOS/5.4/amd64/ lang=us keymap=us ip=dhcp ksdevice=eth0 noipv6 initrd=CentOS/5.4/
    TEXT HELP
    Install CentOS 5.4 (64-bit)
    ENDTXT
LABEL 1
    MENU LABEL CentOS 5.4 (32-bit)
    KERNEL CentOS/5.4/i386/vmlinuz
    APPEND method=nfs:10.10.1.10:/srv/install/CentOS/5.4/i386/ lang=us keymap=us ip=dhcp ksdevice=eth0 noipv6 initrd=CentOS/5.4/
    TEXT HELP
    Install CentOS 5.4 (32-bit)
    ENDTXT
```

Ubuntu

We will now configure support for Ubuntu to be installed.

There are multiple methods to install Ubuntu over the network, however, we will simply boot the Ubuntu Live CD over the network.

For an unattended method for installing Ubuntu over the network, please reference AutomatedNodeDeployment.

Create the directories to store the Ubuntu 9.10 CD.

```
sudo mkdir -p /var/lib/tftpboot/Ubuntu/9.10/i386
sudo mkdir -p /var/lib/tftpboot/Ubuntu/9.10/amd64
sudo mkdir -p /srv/install/Ubuntu/9.10/i386
sudo mkdir -p /srv/install/Ubuntu/9.10/amd64
```

Mount the Ubuntu 9.10 Desktop 64-bit DVD ISO and copy the kernel and initrd to the previously created location (starting around 10.04 a .disk directory on the Ubuntu ISO's are necessary to complete booting, be sure to copy that over as well or booting will freeze).

```
sudo mkdir /mnt/loop
sudo mount -o loop -t iso9660 /location/of/ISO/ubuntu-9.10-desktop-amd64.iso /mnt/loop
sudo cp /mnt/loop/casper/vmlinuz /var/lib/tftpboot/ubuntu/9.10/amd64
sudo cp /mnt/loop/casper/initrd.lz /var/lib/tftpboot/ubuntu/9.10/amd64
sudo cp -R /mnt/loop/* /srv/install/ubuntu/9.10/amd64
sudo cp -R /mnt/loop/.disk /srv/install/ubuntu/9.10/amd64
sudo umount /mnt/loop
```

Mount the Ubuntu 9.10 Desktop 32-bit DVD ISO and copy the kernel and initrd to the previously created location (starting around 10.04 a .disk directory on the Ubuntu ISO's are necessary to complete booting, be sure to copy that over as well or booting will freeze).

```
sudo mkdir /mnt/loop
sudo mount -o loop -t iso9660 /location/of/ISO/ubuntu-9.10-desktop-i386.iso /mnt/loop
sudo cp /mnt/loop/casper/vmlinuz /var/lib/tftpboot/ubuntu/9.10/i386
sudo cp /mnt/loop/casper/initrd.lz /var/lib/tftpboot/ubuntu/9.10/i386
sudo cp -R /mnt/loop/* /srv/install/ubuntu/9.10/i386
sudo cp -R /mnt/loop/.disk /srv/install/ubuntu/9.10/i386
sudo umount /mnt/loop
```

Create the Ubuntu menu.

```
touch /var/lib/tftpboot/Ubuntu/Ubuntu.menu
```

Example /var/lib/tftpboot/Ubuntu/Ubuntu.menu:

```
LABEL 2
    MENU LABEL Ubuntu 9.10 (64-bit)
    KERNEL Ubuntu/9.10/amd64/vmlinuz
    APPEND boot=casper netboot=nfs nfsroot=10.10.1.10:/srv/install/Ubuntu/9.10/amd64 initrd=Ubuntu/9.10/amd64/initrd.lz
    TEXT HELP
```

```

        Boot the Ubuntu 9.10 64-bit DVD
        ENDTEXT
LABEL 1
    MENU LABEL Ubuntu 9.10 (32-bit)
    KERNEL Ubuntu/9.10/i386/vmlinuz
    APPEND boot=casper netboot=nfs nfsroot=10.10.1.10:/srv/install/Ubuntu/9.10/i386 initrd=Ubuntu/9.10/i386/initrd.lz
    TEXT HELP
    Boot the Ubuntu 9.10 32-bit DVD
    ENDTEXT

```

The boot process may appear to halt at "squashfs: version 3.3 (2007/10/31)Phillip Lougher".

Press Alt+Enter and the initialization will resume.

DOS

To support booting DOS via PXE, we will use MEMDISK.

MEMDISK is meant to allow booting legacy operating systems via PXE, and as a workaround for BIOSes where ISOLINUX image support doesn't work.

MEMDISK simulates a disk by claiming a chunk of high memory for the disk and a (very small - 2K typical) chunk of low (DOS) memory for the driver itself, then hooking the INT 13h (disk driver) and INT 15h (memory query) BIOS interrupts.

MEMDISK is an auxillary module used in conjunction with one of the SYSLINUX bootloaders, usually PXELINUX or ISOLINUX. You need a disk image as well as the memdisk file itself. As far as the bootloader is concerned, memdisk is the "kernel" and disk image is the initial ramdisk (initrd).

Copy the MEMDISK module to the root of your TFTP server.

```
sudo cp /usr/lib/syslinux/memdisk /var/lib/tftpboot
```

Create the directory to store DOS.

```
sudo mkdir -p /home/tftpboot/dos/6.22
```

Obtain a MS-DOS bootable floppy disk and create an image from the floppy.

```
sudo dd if=/dev/fd0 of=/home/tftpboot/dos/6.22/floppy.img
```

Create the DOS menu.

```
touch /var/lib/tftpboot/dos/dos.menu
```

Example /var/lib/tftpboot/dos/dos.menu:

```

LABEL 2
    MENU LABEL MS-DOS Floppy Disk
    KERNEL memdisk
    APPEND initrd=dos/msdos622.img
    TEXT HELP
    Boot MS-DOS 6.22
    ENDTEXT

```

The ability to boot DOS via PXE can be extremely useful. You can create a DOS based floppy to flash firmware, or run a multitude of tools.

Utilities

The last example in this article will be booting other utilities via PXE.

Darik's Boot and Nuke ("DBAN") is a self-contained boot disk that securely wipes the hard disks of most computers. DBAN will automatically and completely delete the contents of any hard disk that it can detect, which makes it an appropriate utility for bulk or emergency data destruction.

Create the directories to store DBAN 2.0.

```
sudo mkdir -p /var/lib/tftpboot/utilities/DBAN/2.0/i386
```

Mount the DBAN 2.0 CD and copy the kernel to the previously created location.

```

sudo mount -o loop -t iso9660 /location/of/ISO/dban-beta.2006042900_i386.iso /mnt/loop
sudo cp /mnt/loop/isolinux/dban.bzi /var/lib/tftpboot/DBAN/2.0/i386
sudo umount /mnt/loop

```

Create the utilities menu.

```
touch /var/lib/tftpboot/utilities/utilities.menu
```

Example /var/lib/tftpboot/utilities/utilities.menu:

```

LABEL 18
    MENU LABEL DBAN Boot and Nuke
    KERNEL utilities/dban/dban.bzi
    APPEND nuke="dwipe" silent floppy=0,16,cmos
    TEXT HELP
    Warning - This will erase your hard drive
    ENDTEXT

```

Summary

We have covered the configuration of a system that will allow multiple operating systems to be booted or installed via PXE and some using both PXE and NFS. Many more possibilities exist such as booting:

- SLAX
- Parted Magic
- Knoppix
- xPUD
- VMWare
- openFiler
- RHEL
- etc.

It is also possible to PXE boot the Microsoft Windows installer and make the installation files available using Samba. This process was not covered, however you can review the process at <http://oss.netfarm.it/guides/ris-linux.php>

Troubleshooting

Boot failed: press a key to retry, or wait for reset...

A configuration file was not found and the boot process halts with this error. Check your config file(s). Otherwise, a configuration file is located and the commands within it will be executed (e.g. a boot menu will be displayed and the default option executed when selected).

PXInstallMultiDistro (last edited 2013-06-25 04:13:32 by zeroedout @ localhost[127.0.0.1]:zeroedout)