

Debian PXE Installer Images

Debian, as usual, has their own way of doing things. But, as usual, they are pretty logical when you figure it out.

When you download a Debian *netboot* image, it will be a tar.gz file. Unpacking it into tftp_root on your PXE server sets it up for boot. However, since our goal was to actually have multiple images, we need to do things a little different. I'm assuming you have read [PXE Booting](#) and done all the work there to get things up and running.

We have want to build the tree described in [PXE Booting](#), then download the *netboot* images decompress them, and move the appropriate items into the correct directories. I am lazy so I built this script to do just that, and create some basic menu entries (in default) and add some text to boot.txt.

[getDebian](#)

```
#!/usr/bin/env bash

TFTPDIR=/srv/tftp
MENU=$TFTPDIR/boot.txt
DEFAULTCONFIG=$TFTPDIR/pxelinux.cfg
DEFAULTFILE=$DEFAULTCONFIG/default
MIRROR=ftp://ftp.us.debian.org
DOWNLOADDIR=/tmp
TMPDIR=$DOWNLOADDIR/working

mkdir -p $DEFAULTCONFIG

if [ ! -e $DEFAULTFILE ]
then
    echo 'DISPLAY boot.txt' > $DEFAULTFILE
    echo '' >> $DEFAULTFILE
    echo 'prompt 1' >> $DEFAULTFILE
    echo 'timeout 0' >> $DEFAULTFILE
    echo '' >> $DEFAULTFILE
fi

if [ ! -e $MENU ]
then
    echo 'Available Boot Options:' > $MENU
    echo '===== ' >> $MENU
    echo '' >> $MENU
fi

for arch in i386 amd64
do
    for dist in jessie wheezy
    do
```

```
echo Setting up $dist, $arch
THISDOWNLOAD=$DOWNLOADDIR/$dist/$arch
if [ ! -e $THISDOWNLOAD/netboot.tar.gz ]
then
    mkdir -p $THISDOWNLOAD
    cd $THISDOWNLOAD
    wget -q $MIRROR/debian/dists/$dist/main/installer-
$arch/current/images/netboot/netboot.tar.gz
fi
if [ -e $TEMPDIR ]
then
    rm -fR $TEMPDIR
fi
mkdir -p $TEMPDIR
cd $TEMPDIR
tar -xzf $THISDOWNLOAD/netboot.tar.gz
THISINSTALL=$TFTPDIR/debian/$dist
if [ -e $THISINSTALL ]
then
    rm -fR $THISINSTALL
fi
mkdir -p $THISINSTALL
mv $TEMPDIR/debian-installer/* $THISINSTALL/
# create an entry in default and menu
echo label $dist\_arch\_install >> $DEFAULTFILE
echo "menu label ^Install $dist $arch" >> $DEFAULTFILE
echo "kernel debian/$dist/$arch/linux" >> $DEFAULTFILE
echo "append vga=normal initrd=debian/$dist/$arch/initrd.gz --
quiet" >> $DEFAULTFILE
echo '' >> $DEFAULTFILE
echo $dist\_arch\_install \- \- Install Debian $dist $arch >>
$MENU
done
done
```

The problem with this code is the menu creation. Debian generally has several levels of menu options depending on what you want to do. This will only do the basic *install* option, so if you want more (ie, rescue, install with a gui, etc...), you'll need to create those manually.

Debian appears to use the information found at <http://www.syslinux.org/wiki/index.php?title=PXELINUX>, which discusses pxelinux in detail. This allows for a curses type menuing system, with submenus. However, it results in the menus being stored in numerous files under the boot-screens directory in the installation.

If you only want a few options, without a lot of work, you can enter that directory, find the option you want, then copy/paste the label block into pxelinux.cfg/default (changing the paths, obviously). Otherwise, read the referenced article to build your own menuing system.

NOTE: pxelinux.cfg/ can have many different files in it, and the actions can take place based on

criteria of the machine making the connection, such as MAC address, UUID or IP address (or a portion of it). Again, the documentation at the syslinux web site explains in minute detail. It should be possible, from what I've seen, to modify Debian's menuing tree to be reflected in the PXE boot for multiple images, however.

References

- <http://www.syslinux.org/wiki/index.php?title=PXELINUX>

From:

<https://kb.unixservertech.com/> - **Unix Server Tech Knowledge Base**

Permanent link:

https://kb.unixservertech.com/other/networking/pxebooting/debian_installers

Last update: **2017/05/02 02:24**

