How To Set Up a Diskless Debian GNU/Linux Box

David LaRose, dlr@cs.cmu.edu

June 9, 2006

Version 1.2

Contents

1 Introduction

This file includes instructions for configuring a diskless boot client using Debian GNU/Linux. The resulting files can be served from a tftp server to network boot diskless boxes. Unlike many of the available diskless boot images, this one doesn't mount its root partition from a remote server. Instead, the entire root partition is loaded from the initial boot image. This means means that the initial ramdisk is very large. If you add much to this image, it can easily get too large to serve via tftp. One solution is to boot a small ramdisk, and then use http or ftp to download a larger image. This doesn't include instructions for doing that.

2 Credits

The material in this document draws heavily from Dave Vehrs' "Installing Debian onto USB flash media with everything encrypted," webpage. This webpage is currently available at http://www.debian-administration.org/articles/1
Additional information was drawn from Thomas Hood's readonly-root-README.txt file, currently available at http://panopticon.csustan.edu/thood/readonly-root-README.txt. Please assume that any good ideas in the following text came from one of these two sources.

3 Build the Boot Image

3.1 Initial Setup

Make a basic Debian install. I've used /root/diskImages as the working directory, but you might prefer to work in different partition, such as /home or /var.

```
workstation:~/diskImages# apt-get update
workstation:~/diskImages# apt-get install debootstrap
workstation:~/diskImages# mkdir tmpRoot
workstation:~/diskImages# /usr/sbin/debootstrap --arch i386 sid tmpRoot
```

Check out what's in the newly created dev directory, and copy any extra devices you expect to need.

```
workstation:~/diskImages# cp -a /dev/sd* tmpRoot/dev/
workstation:~/diskImages# cp -a /dev/tty[1-6] tmpRoot/dev/
```

3.2 Run-of-the-mill Configuration

Chroot to the new system and set up any users. For the rest of this document, we'll use the prompt "chroot:" to indicate that that we're in a chrooted environment. In real life, you won't get this prompt, so you'll have to be careful to keep track of whether or not you're chrooted. This is important because if you think you're in the chroot environment, but you're actually not, you might accidentally start to configure your workstation for diskless boot.

```
workstation:~/diskImages# chroot /root/diskImages/tmpRoot/ \
                          /bin/su -
chroot:~# dpkg-reconfigure passwd
chroot:~# passwd root
chroot:~# adduser dlr
```

Build an fstab. Note that the root filesystem is commented out here. This is a bit of a hack, but seems to work fine.

```
chroot:~# mkdir /var/lib/exim4 /var/spool/exim4
chroot:~# cat > /etc/fstab << EOF
#/etc/fstab: static file system information.
# /dev/root  /               ext2    defaults          0 0
proc       /proc            proc    defaults          0 0
sysfs      /sys             sysfs   defaults          0 0
tmpfs      /etc/network/run tmpfs   defaults,noatime  0 0
```

```
tmpfs      /tmp             tmpfs   defaults,noatime  0 0
tmpfs      /var/lock        tmpfs   defaults,noatime  0 0
tmpfs      /var/log         tmpfs   defaults,noatime  0 0
tmpfs      /var/mail        tmpfs   defaults,noatime  0 0
tmpfs      /var/run         tmpfs   defaults,noatime  0 0
tmpfs      /var/spool/cron  tmpfs   defaults,noatime  0 0
tmpfs      /var/tmp         tmpfs   defaults,noatime  0 0
tmpfs      /var/lib/exim4   tmpfs   defaults,noatime  0 0
tmpfs      /var/spool/exim4 tmpfs   defaults,noatime  0 0
EOF
```

Install whatever software packages you expect to need. We temporarily add a dummy entry to /etc/fstab, mirroring the entry for the root filesystem of the machine we're working on. We do this because the stock mkinitrd script gets confused if we don't have a root filesystem listed. Note that the resulting initrd is incorrect, but we don't care since we're going to build our own initrd anyway.

```
chroot:~# mount -a
chroot:~# echo "/dev/sda1 / ext2 defaults 0 0" >> /etc/fstab
chroot:~# echo "deb ftp://ftp.us.debian.org/debian/ unstable main" \
          > /etc/apt/sources.list
chroot:~# apt-get update
chroot:~# apt-get install linux-image-2.6-386
chroot:~# apt-get install discover1 libdiscover1
chroot:~# apt-get install mailx
chroot:~# apt-get install resolvconf
chroot:~# apt-get clean
chroot:~# tzconfig
chroot:~# cp /etc/fstab /etc/fstab.broken
chroot:~# head --lines=-1 < /etc/fstab.broken > /etc/fstab
chroot:~# /etc/init.d/exim4 stop
chroot:~# umount -a
chroot:~# umount /proc
```

3.3 Remove Unneeded Files

At this point, you may want to remove unneeded packages. This will reduce the size of the root filesystem. While doing this, it may be helpful to remember the –force-depends option to dpkg.

```
chroot:~# dpkg --purge --force-depends perl \
          libparse-recdescent-perl yaird perl-modules
chroot:~# rm -rf /usr/share/doc/*
```

You might also want to check out the localepurge package, which can significantly shrink the image. I haven't needed to do this yet, so the following line is untested.

```
chroot:~# apt-get install localepurge
```

3.4 Customizations for Read-Only Root

Customize the system a little to allow running with a read-only root filesystem. Thanks to Dave Vehrs and Thomas Hood for these ideas.

```
chroot:~# rm mtab
chroot:~# ln -s /proc/mounts /etc/mtab
chroot:~# rm -f /etc/blkid.tab*
chroot:~# ln -s /dev/null /etc/blkid.tab
chroot:~# cp /etc/default/rcS /etc/default/rcS.orig
chroot:~# sed -e s/EDITMOTD=yes/EDITMOTD=no/ \
          < /etc/default/rcS.orig > /etc/default/rcS
chroot:~# rm -rf /etc/resolvconf/run
chroot:~# ln -s /dev/shm/resolvconf /etc/resolvconf/run
```

3.5 Network Related Configuration

Configure the network.

```
chroot:~# echo "127.0.0.1 localhost.localdomain localhost" \
          > /etc/hosts
chroot:~# echo "auto eth0" >> /etc/network/interfaces
chroot:~# echo "iface eth0 inet dhcp" \
          >> /etc/network/interfaces
```

Add provision to set the hostname from dhcp. I suspect there's a better way to do this, but I don't know it.

```
chroot:~# rm /etc/hostname
chroot:~# ln -s /var/run/hostname /etc/hostname
chroot:~# ln -s ../init.d/dlr_sethostname \
          /etc/rcS.d/S41dlr_sethostname
chroot:~# cat > /etc/init.d/dlr_sethostname << EOF
#! /bin/sh
PATH=/sbin:/bin:/usr/sbin:/usr/bin
set_hostname() {
  recoveredName="\`grep 'host-name ' /var/run/dhclient.eth0.leases \\
    | sed -e 's/^.*host-name \"//' -e 's/\"\;$//' \\
    | tail -1\`"
  if [ -z \$recoveredName ]; then
    recoveredName="anonymous"
  fi
  echo \$recoveredName > /var/run/hostname
}
case "\$1" in
  start)
    echo "Recovering hostname from dhcp lease..."
    set_hostname
    if [ -f /var/run/hostname ]; then
      cat /var/run/hostname
```

```
      /etc/init.d/hostname.sh start
    else
      echo "Hostname recovery failed."
    fi
    ;;
  *)
    ;;
esac
: exit 0
EOF
chroot:~# chmod a+x /etc/init.d/dlr_sethostname
```

Add provision to set the mailname from dhcp. This uses the same trick that we used for setting the hostname.

```
chroot:~# rm /etc/mailname
chroot:~# ln -s /var/run/mailname /etc/mailname
chroot:~# ln -s ../init.d/dlr_setmailname \
          /etc/rcS.d/S41dlr_setmailname
chroot:~# cat > /etc/init.d/dlr_setmailname << EOF
#! /bin/sh
PATH=/sbin:/bin:/usr/sbin:/usr/bin
set_mailname() {
  recoveredName="\`grep 'host-name ' /var/run/dhclient.eth0.leases \\
    | sed -e 's/^.*host-name \"//' -e 's/\"\;$//' \\
    | tail -1\`"
  if [ -z \$recoveredName ]; then
    recoveredName="anonymous"
  fi
  recoveredDomain="\`grep 'domain-name ' \\
                       /var/run/dhclient.eth0.leases \\
    | sed -e 's/^.*domain-name \"//' -e 's/\"\;$//' \\
    | tail -1\`"
  if [ -z \$recoveredDomain ]; then
    recoveredDomain="nodomain.notld"
  fi
  echo \$recoveredName.\$recoveredDomain > /var/run/mailname
}
case "\$1" in
  start)
    echo "Recovering mailname from dhcp lease..."
    set_mailname
    if [ -f /var/run/mailname ]; then
      cat /var/run/mailname
    else
      echo "Mailname recovery failed."
    fi
    ;;
  *)
```

```
    ;;
esac
: exit 0
EOF
chroot:~# chmod a+x /etc/init.d/dlr_setmailname
```

3.6 Final Steps

Specify which modules should be loaded at boot time. Make sure you include the appropriate module for your network card.

```
chroot:~# echo "via-rhine" >> /etc/modules
```

Uncomment the root filesystem in /etc/fstab, save both versions of /etc/fstab, and return to the un-chrooted environment.

```
chroot:~# cp /etc/fstab /etc/fstab.noRoot
chroot:~# sed -e "s/# \/dev\/root/\/dev\/root/" \
          < /etc/fstab.noRoot > /etc/fstab
chroot:~# cp /etc/fstab /etc/fstab.withRoot
chroot:~# logout
```

Build the ramdisk.

```
workstation:~/diskImages# mkcramfs tmpRoot \
                      initrd.img-2.6.14-1-386-netboot-1.0
```

4 Configure the Boot Server

4.1 Install Necessary Packages

```
server:~# apt-get install dhcp3-server
server:~# apt-get install atftpd
server:~# apt-get install syslinux
server:~# apt-get clean
```

4.2 Configure tftp

The defaults provided by debconf are generally OK. For security reasons, it probably makes sense to configure tftpd to be started by inetd, and use hosts.{allow,deny} to control access. Here's a sample line you can add to /etc/hosts.allow:

```
in.tftpd: 192.168.1.*
```

You'll need to restart inetd in order for the change to take effect.

```
server:~# /etc/init.d/inetd restart
```

Additionally, you'll need to make the boot files available through tftp as follows:

```
server:~# mkdir /tftpboot
server:~# cp /usr/lib/syslinux/pxelinux.0 /tftpboot
server:~# cp vmlinuz-2.6.14-1-386 /tftpboot/vmlinuz
server:~# cp initrd.img-2.6.14-1-386-netboot-1.0 \
          /tftpboot/initrd.img
server:~# mkdir /tftpboot/pxelinux.cfg
server:~# cat > /tftpboot/pxelinux.cfg/default << EOF
DEFAULT vmlinuz initrd=initrd.img ramdisk=200000 root=/dev/ram auto
TIMEOUT 50
EOF
```

4.3 Configure dhcp

1. Edit /etc/default/dhcp3-server to specify the correct network interface.
2. Edit /etc/dhcp3/dhcpd.conf to match your expected hosts. Here's an example you can start from, if you like. Please also see the default version of the file (the one that's provided when you install dhcp3-server) since it has lots of nice comments.

```
server:~# cat > /etc/dhcp3/dhcpd.conf << EOF
# dhcp3 server configuration file.
ddns-update-style none;
# option definitions common to all supported networks.
option domain-name "mynetwork.com";
option domain-name-servers 192.168.1.1;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.1.255;
option routers 192.168.1.1;
default-lease-time 600;
max-lease-time 7200;
# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be specified.
authoritative;
# Use this to send dhcp log messages to a different log file (you
# also have to hack syslog.conf to complete the redirection).
log-facility local7;
# Subnet definition
subnet 192.168.1.0 netmask 255.255.255.0 {
  range 192.168.1.1 192.168.1.255;
}
# Entries for specific boxes.
host svn0 {
  hardware ethernet 00:41:64:de:a2:b0;
  next-server 192.168.1.1;
  filename "/pxelinux.0";
  fixed-address 192.168.1.80;
  option host-name "svn0";
}
EOF
```

3. Start the dhcp server:

```
sever:~# /etc/init.d/dhcp3-server start
```

5 Configure the Client Machine

The diskless machines have very little state, so there's not much to configure. All there is to do is edit the BIOS setup so that it allows network booting. On, for example, Supermicro PDSLA motherboards with BIOS rev1.0a, this involves two settings:

1. Advanced->PnP/PCI Configurations->Onboard Lan Boot ROM: Enabled
2. Boot->Hard Disk Boot Priority->First Boot Device: LAN

6 Updating the Image

At some point, you may want to update the image. You can do so as follows. Note that if you're going to install a kernel, you'll have to do the "fake root partition trick" from section 3.2 again.

```
workstation:~/diskImages# chroot /root/diskImages/tmpRoot/ \
                          /bin/su -
workstation:~# cp /etc/fstab.noRoot /etc/fstab
workstation:~# mount -a
workstation:~# apt-get update
workstation:~# apt-get upgrade
workstation:~# apt-get clean
workstation:~# umount -a
workstation:~# cp /etc/fstab.withRoot /etc/fstab
workstation:~# logout
```

7 Closing

Please send suggestions and improvements to David LaRose, dlr@cs.cmu.edu.