PXE

From **Wikipedia:Preboot Execution Environment**:

> *The Preboot eXecution Environment (PXE, also known as Pre-Execution Environment; sometimes pronounced "pixie") is an environment to boot computers using a network interface independently of data storage devices (like hard disks) or installed operating systems.*

Related articles

Diskless system

In this guide, PXE is used to boot the installation media with an appropriate option-rom that supports PXE on the target. This works well when you already have a server set up.

Contents

Preparation

Get the latest official install media from **download page (https://www.archlinux.org/download/)**.

Next mount the image:

```
# mkdir -p /mnt/archiso
# mount -o loop,ro archlinux-2017.12.01-x86_64.iso /mnt/archiso
```

Server setup

You will need to setup a DHCP, **TFTP**, and **HTTP server** to configure networking, load pxelinux/kernel/initramfs, and finally load the root filesystem (respectively).

Arch currently only supports BIOS-style PXE booting. See **FS#50188 (https://bugs.archlinux.org/task/50188)** for more information.

Network

Bring up your wired NIC, and assign it an address appropriately.

```
# ip link set eth0 up
# ip addr add 192.168.0.1/24 dev eth0
```

DHCP + TFTP

You will need both a DHCP and TFTP server to configure networking on the install target and to facilitate the transfer of files between the PXE server and client; **dnsmasq** does both, and is extremely easy to set up.

Install the **`dnsmasq (https://www.archlinux.org/packages/?name=dnsmasq)`** package.

Configure *dnsmasq*:

```
# /etc/dnsmasq.conf
```

```
port=0
interface=eth0
bind-interfaces
dhcp-range=192.168.0.50,192.168.0.150,12h
dhcp-boot=/arch/boot/syslinux/lpxelinux.0
dhcp-option-force=209,boot/syslinux/archiso.cfg
dhcp-option-force=210,/arch/
dhcp-option-force=66,192.168.0.1
enable-tftp
tftp-root=/mnt/archiso
```

Start `dnsmasq.service`.

HTTP

Thanks to recent changes in **archiso**, it is now possible to boot from HTTP (`archiso_pxe_http` initcpio hook) or NFS (`archiso_pxe_nfs` initcpio hook); among all alternatives, darkhttpd is by far the most trivial to setup (and the lightest-weight).

First, **install** the **`darkhttpd (https://www.archlinux.org/packages/?name=darkhttpd)`** package.

Then start *darkhttpd* using our `/mnt/archiso` as the document root:

```
# darkhttpd /mnt/archiso
```

```
darkhttpd/1.8, copyright (c) 2003-2011 Emil Mikulic.
listening on: http://0.0.0.0:80/
```

Note that it is important that the server is running on port `80`. If you start *darkhttpd* without root access it will default to `8080`. The client will try to access port 80 and the boot will fail.

Installation

For this portion you will need to figure out how to tell the client to attempt a PXE boot; in the corner of the screen along with the normal post messages, usually there will be some hint on which key to press to try PXE booting first. On an IBM x3650 `F12` brings up a boot menu, the first option of which is *Network*; on a Dell PE 1950/2950 pressing `F12` initiates PXE booting directly.

Boot

Looking at **journald** on the PXE server will provide some additional insight to what exactly is going on during the early stages of the PXE boot process:

```
# journalctl -u dnsmasq.service -f
```

```
dnsmasq-dhcp[2544]: DHCPDISCOVER(eth1) 00:1a:64:6a:a2:4d
dnsmasq-dhcp[2544]: DHCPOFFER(eth1) 192.168.0.110 00:1a:64:6a:a2:4d
dnsmasq-dhcp[2544]: DHCPREQUEST(eth1) 192.168.0.110 00:1a:64:6a:a2:4d
dnsmasq-dhcp[2544]: DHCPACK(eth1) 192.168.0.110 00:1a:64:6a:a2:4d
dnsmasq-tftp[2544]: sent /mnt/archiso/arch/boot/syslinux/pxelinux.0 to 192.168.0.110
dnsmasq-tftp[2544]: sent /mnt/archiso/arch/boot/syslinux/archiso.cfg to 192.168.0.110
dnsmasq-tftp[2544]: sent /mnt/archiso/arch/boot/syslinux/whichsys.c32 to 192.168.0.110
dnsmasq-tftp[2544]: sent /mnt/archiso/arch/boot/syslinux/archiso_pxe_choose.cfg to 192.168.0.110
dnsmasq-tftp[2544]: sent /mnt/archiso/arch/boot/syslinux/ifcpu64.c32 to 192.168.0.110
dnsmasq-tftp[2544]: sent /mnt/archiso/arch/boot/syslinux/archiso_pxe_both_inc.cfg to 192.168.0.110
dnsmasq-tftp[2544]: sent /mnt/archiso/arch/boot/syslinux/archiso_head.cfg to 192.168.0.110
dnsmasq-tftp[2544]: sent /mnt/archiso/arch/boot/syslinux/archiso_pxe32.cfg to 192.168.0.110
dnsmasq-tftp[2544]: sent /mnt/archiso/arch/boot/syslinux/archiso_pxe64.cfg to 192.168.0.110
```

```
dnsmasq-tftp[2544]: sent /mnt/archiso/arch/boot/syslinux/archiso_tail.cfg to 192.168.0.110
dnsmasq-tftp[2544]: sent /mnt/archiso/arch/boot/syslinux/vesamenu.c32 to 192.168.0.110
dnsmasq-tftp[2544]: sent /mnt/archiso/arch/boot/syslinux/splash.png to 192.168.0.110
```

After you load `pxelinux.0` and `archiso.cfg` via TFTP, you will (hopefully) be presented with a **syslinux** boot menu with several options, where you can select *Boot Arch Linux (x86_64) (HTTP)*.

Next the kernel and initramfs (appropriate for the architecture you selected) will be transferred, again via TFTP:

```
dnsmasq-tftp[2544]: sent /mnt/archiso/arch/boot/x86_64/vmlinuz to 192.168.0.110
dnsmasq-tftp[2544]: sent /mnt/archiso/arch/boot/x86_64/archiso.img to 192.168.0.110
```

If all goes well, you should then see activity on darkhttpd coming from the PXE-target; at this point the kernel would be loaded on the PXE-target, and in init:

```
1348347586 192.168.0.110 "GET /arch/aitab" 200 678 "" "curl/7.27.0"
1348347587 192.168.0.110 "GET /arch/x86_64/root-image.fs.sfs" 200 107860206 "" "curl/7.27.0"
1348347588 192.168.0.110 "GET /arch/x86_64/usr-lib-modules.fs.sfs" 200 36819181 "" "curl/7.27.0"
1348347588 192.168.0.110 "GET /arch/any/usr-share.fs.sfs" 200 63693037 "" "curl/7.27.0"
```

After the root filesystem is downloaded via HTTP, you will eventually end up at the normal live system root **zsh** prompt.

Post-boot

Unless you want all traffic to be routed through your PXE server (which will not work anyway unless you **set it up properly**), you will want to **stop** `dnsmasq.service` and get a new lease on the install target, as appropriate for your network layout.

You can also kill *darkhttpd*; the target has already downloaded the root filesystem, so it is no longer needed. While you are at it, you can also unmount the installation image:

```
# umount /mnt/archiso
```

At this point you can follow the **Installation guide**.

Alternate Methods

As implied in the syslinux menu, there are several other alternatives:

NFS

You will need to set up an **NFS server** with an export at the root of your mounted installation media, which would be `/mnt/archiso` if you followed **#Preparation**. After setting up the server, add the following line to your `/etc/exports` file:

```
/etc/exports
```

```
/mnt/archiso 192.168.0.0/24(ro,no_subtree_check)
```

If the server was already running, re-export the filesystems with `exportfs -r -a -v`.

The default settings in the installer expect to find the NFS at `/run/archiso/bootmnt`, so you will need to edit the boot options. To do this, press Tab on the appropriate boot menu choice and edit the `archiso_nfs_srv` option accordingly:

```
archiso_nfs_srv=${pxeserver}:/mnt/archiso
```

Alternatively, you can use `/run/archiso/bootmnt` for the entire process.

After the kernel loads, the Arch bootstrap image will copy the root filesystem via NFS to the booting host. This can take a little while. Once this completes, you should have a running system.

NBD

Install the **nbd (https://www.archlinux.org/packages/?name=nbd)** package and configure it:

/etc/nbd-server/config

```
[generic]
[archiso]
    readonly = true
    exportname = /srv/archlinux-2017.12.01-x86_64.iso
```

Start `nbd.service`.

Existing PXE Server

If you have an existing PXE server with a **PXELINUX** system setup (e.g. a combination of DHCP and **TFTP**), you can add the following menu items to your `pxelinux.cfg` file in order to boot Arch via your preferred method:

```
LABEL 2
        MENU LABEL Arch Linux x86_64
        LINUX /path/to/extracted/Arch/ISO/arch/boot/x86_64/vmlinuz
        INITRD /path/to/extracted/Arch/ISO/arch/boot/intel_ucode.img,/path/to/extracted/Arch/ISO/arch/boot/x86_64/archiso.img
        APPEND archisobasedir=arch archiso_http_srv=http://httpserver/path/to/extracted/Arch/ISO/ ip=::
        SYSAPPEND 2
        TEXT HELP
        Arch Linux 2017.12.01 x86_64
        ENDTEXT
```

You can replace `archiso_http_srv` with `archiso_nfs_srv` for NFS or `archiso_nbd_srv` for NBD. Adding the `ip=` instruction is necessary to instruct the kernel to bring up the network interface before it attempts to mount the installation medium over the network. See **README.bootparams (https://git.archlinux.org/archiso.git/plain/docs/README.bootparams)** for available boot parameters.

DHCP interface rename bug

FS#36749 (https://bugs.archlinux.org/task/36749) causes default **predictable network interface renaming (http://www.freedesktop.org/wiki/Software/systemd/PredictableNetworkInterfaceNames/)** to fail and then dhcp client to fail because of it. A workaround is to add the kernel boot parameter `net.ifnames=0` to disable predictable interface names.

Low memory systems

The `copytoram` **initramfs** option can be used to control whether the root filesystem should be copied to ram in its entirety in early-boot.

It highly recommended to leave this option alone, and should only be disabled if entirely necessary (systems with less than ~256MB physical memory). Append `copytoram=n` to your kernel line if you wish to do so.

Note: As this requires loop-mounting squashfs from a mounted remote filesystem, `copytoram=n` and `archiso_pxe_http` are mutually exclusive.

Retrieved from "https://wiki.archlinux.org/index.php?title=PXE&oldid=507511"

- This page was last edited on 14 January 2018, at 09:54.