Installing the PXE Server

This tutorial was done with a Debian 8. It will only be possible to automatically boot an OS according to the module's MAC address. In this configuration there is no GUI to select a boot image.

All operations where done with root permissions.

First the following services have to be installed:

```
apt-get install dnsmasq nfs-kernel-server
```

`dnsmasq` serves as the DHCP Server to tell the clients what IP address to use and which kernel they have to load. Also it runs the TFTP Server that provides the kernel images.

`nfs-kernel-server` provides the NFS network share for the RootFS and a persistent storage.

Setting up dnsmasq

The config file can be opened with

```
nano /etc/dnsmasq.conf
```

All DHCP Options used in the following config can be found here Link The following shows an example configuration for 3 different modules and 4 different kernels.

```
#Binds the DHCP server to eth0
interface=eth0
#Sets the IP range with a lease time of 8 hours
dhcp-range=192.168.13.201,192.168.13.249,255.255.255.0,8h

#Enables the TFTP service of dnsmasq
enable-tftp
#Bind the TFTP root folder  to the set folder
tftp-root=/srv/tftpboot/
#DHCP Option 66 sets which TFTP Server will be used by the client
dhcp-option=66,192.168.13.80
#DHCP Option 42 sets a NTP time server
dhcp-option=42,130.133.1.10


#The following sets a flag for all MAC addresses starting with the given
number,
#so it is possible to set different setups for different modules
dhcp-host=00:14:2d:*:*:*,set:toradexApalis
dhcp-host=00:0e:c6:*:*:*,set:toradexColibriAndroid
#"set:christmannExynosLinux" has to be changed to
"set:christmannExynosAndroid" in order to load
```

```
#the Android kernel
dhcp-host=70:b3:d5:56:*:*,set:christmannExynosLinux


#These different tags can be used with "tag:"tagname","option""
#"dhcp-boot" defines which kernel will be loaded from the TFTP root path
#"dhcp-option" gives an option to the set MAC address group

#Kernel filename for Toradex Apalis T30 provided by the TFTP Server
dhcp-boot=tag:toradexApalis,toradexApalis_uImage
#Root path that is shared by the nfs-kernel-server
dhcp-option=tag:toradexApalis,17,/srv/nfs/apalisT30/rootfs/

#Kernel filename for Toradex Colibri T20
dhcp-boot=tag:toradexColibri,toradexColibri_uImage
#Root path
dhcp-option=tag:toradexColibri,17,/srv/nfs/colibriT20/linux/rootfs/

#Kernel filename for Christmann Apalis Exynos Linux
dhcp-boot=tag:christmannExynosLinux,uImage-ExynosLinux
#Root path
dhcp-option=tag:christmannExynosLinux,17,/srv/nfs/apalisExynos/linux/rootfs/

#Kernel filename for Christmann Apalis Exynos Android
dhcp-boot=tag:christmannExynosAndroid,uImage-ExynosAndroid
#Root path
dhcp-
option=tag:christmannExynosAndroid,17,/srv/nfs/apalisExynos/android/rootfs
```

After the `dnsmasq.conf` has been changed, the dnsmasq service has to be restarted in order to apply the changes using:

```
/etc/init.d/dnsmasq restart
```

Then the `uImage` files must be copied to `/srv/tftpboot/*` with the name that was defined in `dhcp-boot`.

Setting up nfs-kernel-server

The RootFS and/or a persistent storage are exported to the client via NFS. This is the location where the root filesystem of the desired distribution must be copied to.

The shared/exported folders can be edited in this file:

```
nano /etc/exports
```

Here is an example configuration:

```
/srv/nfs/apalisT30/rootfs       *(rw,no_root_squash,no_subtree_check)
/srv/nfs/storage/               *(rw,async,no_root_squash)
/srv/nfs/colibriT20             *(rw,no_root_squash,no_subtree_check)
/srv/nfs/apalisExynos/          *(ro,no_root_squash,no_subtree_check)
```

First in line is the folder that will be shared via NFS.
The * in front of the bracket sets the clients that are allowed to access the share. In this case everyone can access the share, but also singe IP addresses or ranges can be set.
The Arguments in the brackets are the following:

`rw`	Read Write access
`ro`	Read Only access
`async`	Better performance with the danger of data loss if the server crashes or is shut down. Default is `sync`
`no_root_squash`	Enables access to files that are owned by root on the server
`no_subtree_check`	disables the subtree checking when accessing a file, this in on by default

A detailed description can be found at Ubuntuusers or with `man exports`

To apply the changes the `nfs-kernel-server` has to be restarted.

```
/etc/init.d/nfs-kernel-server restart
```