

Setting up a Linux/Unix Server for the dBox

Table of contents

1 Yet another guide on setting up a Linux server?.....	2
2 General.....	2
2.1 YADD and its advantages.....	2
2.2 Host name resolving.....	3
3 The DHCP Server.....	3
3.1 Troubleshooting.....	4
4 The TFTP Server.....	5
4.1 Troubleshooting.....	5
5 The NFS Server.....	5
5.1 Troubleshooting.....	6
6 Other possible services.....	6
6.1 DNS Name server.....	6
6.2 RARP server, Bootp server.....	6
7 Flashing.....	6
8 Troubleshooting.....	7
9 Links.....	7

1. Yet another guide on setting up a Linux server?

The goal of this guide is to provide help on setting up a Unix or Linux server for providing the dBox with some useful services. It is aimed at the beginner to intermediate user. Nothing really new is presented here. The emphasis is on traditional Unix services, getting them to work, to troubleshoot them.

2. General

It is only when you network the dBox, that you can take advantage of its full power. A number of services exist, and new are invented regularly. Some of these are multimedia related, and enables the box to digitally record DVB-transmissions, or to reproduce video or audio material. This is not the topic for the present guide, that instead focuses on the more traditional Unix services.

The three services we will cover are:

NFS

Provides file server service

TFTP

Provides the client with a file to boot

DHCP

Provides the client with parameters, that it needs for booting

Typically, but not by necessity, they reside on the same host.

It is sometimes sensible to set up only, e.g., an NFS-Server.

If using newmake to compile the CDK, outlines for `/etc/hosts`, `/etc/exports`, and `/etc/dhcpd.conf` are generated, by the target `server` support.

2.1. YADD and its advantages

It is possible for the dBox (in debug mode!) to boot entirely from a server, not using any (nonvolatile) internal memory. The necessary parameters for booting will be received from a server, another (possibly different) server provides it with the necessary boot loader (u-boot, in the past "ppcboot") and secondly with a Linux kernel. The root file system is NFS-mounted from a server. This mode of operation is called YADD ("Yet Another Dbox Distribution"), or CDK ("Cross Development Kit"), an extraordinarily silly, but established, name. This mode of operation has many advantages:

- The flash memory, i.e. "its" operating system, is not affected in any way.
- Using the flash functions of, e.g. Neutrino, the flash can be read out, or the flash can be new programmed.
- Disk space, also in the root partition, is "unlimited".
- Very practical for development: After a successful build, the changed software immediately resides on the dBox, without need for flashing.

2.2. Host name resolving

In many situations, computers have to be referred to in a unique fashion. This can be done with IP-Numbers, or, if the involved computers can understand it, symbolic names. To translate the symbolic names to IP-numbers ("hostname resolving"), a number of mechanisms have been proposed. For home networks, usually the file `/etc/hosts` is used. For a small home network, this may look like:

```
127.0.0.1    localhost
192.168.1.1  workcomputer
192.168.1.2  server
192.168.1.5  dbox nokia
192.168.1.6  sagem
```

This file is then distributed to all computers in the network. Now the symbolic names can be used in configuration files.

3. The DHCP Server

The `dhcpd`-daemon supplied with most Linux distribution is a very versatile program. It implements, not only the [DHCP](#)-protocol, but also the BOOTP protocol. It provides a client, providing only its MAC-address, with the information necessary to boot from a TFTP- and NFS-server. I recommend not to configure it using the system administration tool, but instead to generate a suitable configuration file `/etc/dhcpd.conf`. If using `newmake` to compile CDK, a suitable `/etc/dhcpd.conf` will be generated by the target `serversupport`, adapted to the parameter used when running the `configure`-command. This file will typically look like:

```
# This is a template for dhcp.conf
# Copy to the appropriate location for your server,
# typically /etc/dhcp.conf.
# You may have to modify this file manually.
# Please see the documentation for your server and for dhcpd.

ddns-update-style none;
subnet 192.168.1.0 netmask 255.255.255.0 {
}

host dbox {
    fixed-address 192.168.1.5;
    hardware ethernet 00:50:9c:xx:xx:x;
    allow bootp;
    server-name "godzilla";
    next-server "godzilla";
    option root-path "/tuxbox/cdkroot";
    if exists vendor-class-identifier {
        filename "kernel-cdk";
    } else {
        filename "u-boot";
    }
}
```

where "godzilla" is the name of the DHCP-, TFTP-, and NFS-server.

The MAC-address of the dBox needs to be entered here. Often, it is printed on the back plate of the dBox. Otherwise, a command like `arp -a` can often be used to determine it. The MAC-number of a dBox always starts with "00:50:9C".

As of CVS-Stand 2006-02-01, the root path can be entered in the form `IP-Address:/directory`, not forcing TFTP- and NFS-server to coincide. However, using IP-Name instead of IP-Address is not possible.

The `dhcpd-daemon` can be started either from the system administration tool, or directly with `/etc/init.d/dhcpd start`.

For more information, see the man-pages for `dhcpd` and `dhcpd.conf`, or the source of the software, [ISC](#).

Unfortunately, unless you remove the Ethernet cable from the dBox when booting, unlike the NFS-server and the TFTP-server, the `dhcpd-daemon` interferes with normal operation of the dBox -- it will simply force the box to boot from the server. I do not know of a really elegant solution: personally I let `dhcpd` be off by default, and start it, using `/etc/init.d/dhcpd start`, when needed.

3.1. Troubleshooting

As can be inferred from the configuration file, there is a fair number of things that can go wrong. Also, as is discussed in [this thread](#), some versions of `dhcpd` do not work with the Tuxbox. For troubleshooting, use the system log, typically `/var/log/messages`. For reference, we show a successful boot ("godzilla" is the server, "dbox" the dBox, having MAC-address 00:50:9c:xx:xx:xx):

```
Jan 23 14:28:21 godzilla dhcpd: BOOTREQUEST from 00:50:9c:xx:xx:xx via eth0
Jan 23 14:28:21 godzilla dhcpd: BOOTREPLY for 192.168.1.5 to dbox (00:50:9c:xx:xx:xx) via eth0
Jan 23 14:28:22 godzilla dhcpd: DHCPDISCOVER from 00:50:9c:xx:xx:xx via eth0
Jan 23 14:28:22 godzilla dhcpd: DHCPOFFER on 192.168.1.5 to 00:50:9c:xx:xx:xx via eth0
Jan 23 14:28:22 godzilla dhcpd: DHCPREQUEST for 192.168.1.5 (192.168.1.1) from 00:50:9c:xx:xx:xx via eth0
Jan 23 14:28:22 godzilla dhcpd: DHCPACK on 192.168.1.5 to 00:50:9c:xx:xx:xx via eth0
Jan 23 14:28:23 godzilla dhcpd: DHCPDISCOVER from 00:50:9c:xx:xx:xx via eth0
Jan 23 14:28:23 godzilla dhcpd: DHCPOFFER on 192.168.1.5 to 00:50:9c:xx:xx:xx via eth0
Jan 23 14:28:23 godzilla dhcpd: DHCPREQUEST for 192.168.1.5 (192.168.1.1) from 00:50:9c:xx:xx:xx via eth0
Jan 23 14:28:23 godzilla dhcpd: DHCPACK on 192.168.1.5 to 00:50:9c:xx:xx:xx via eth0
Jan 23 14:28:27 godzilla dhcpd: DHCPDISCOVER from 00:50:9c:xx:xx:xx via eth0
Jan 23 14:28:27 godzilla dhcpd: DHCPOFFER on 192.168.1.5 to 00:50:9c:xx:xx:xx via eth0
```

```
Jan 23 14:28:27 godzilla dhcpd: DHCPREQUEST for 192.168.1.5
(192.168.1.1) from 00:50:9c:xx:xx:xx via eth0
Jan 23 14:28:27 godzilla dhcpd: DHCPACK on 192.168.1.5 to
00:50:9c:xx:xx:xx via eth0
Jan 23 14:28:37 godzilla rpc.mountd: authenticated mount request from
dbox:800 for /tuxbox/cdkroot (/tuxbox/cdkroot)
```

The last line is not from dhcpd, but shows that the NFS-mount of the root succeeded.

4. The TFTP Server

[TFTP](#) is a very simplified file transfer protocol. For the Tuxbox, it serves the client with the boot loader `u-boot` and the Linux kernel `kernel-cdk`, as well as (optionally) the logo files `logo-fb` and `logo-lcd`. The TFTP-Service is normally set up using the system's administration tool. The only real interesting setup option is its base directory (called "Boot image directory" in YAST2/SuSE). This is where `u-boot` and `kernel-cdk` reside. It should normally be the same as the `--with-bootdir` parameter to `configure`.

4.1. Troubleshooting

There is not too much that can go wrong, basically only "File not found". These messages can often be found in the system log, typically `/var/log/messages`.

Since the TFTP-daemon "chroot"-s to its base directory, all file names are interpreted with respect to this directory, which can sometimes be confusing.

5. The NFS Server

The [Network file system \(NFS\)](#) is the most common remote file system in the Unix world. For a file server for the dBox, there appears to be a consensus that it offers better performance and reliability than other alternatives.

The NFS server is turned on using your standard system administration tool, like YAST2 in SuSE. For the configuration (what file systems are exported, to what client, and with what properties), it really all boils down to editing the file `/etc/exports`, although system administration tools may offer "user friendly" ways of editing this file.

Typically, the file may look like:

```
/tuxbox/cdkroot      dbox(rw,sync,no_root_squash)
/multimedia/music   dbox(ro)
/multimedia/streaming dbox(rw)
```

This file says that the directory `/tuxbox/cdkroot` (on the server) is exported to the host with the name "dbox" (as found in `/etc/exports`), the host may mount it with write-access. The options "sync" and "no_root_squash" should be used for the file system to be mounted as root, otherwise it is not needed. The second line states that the filesystem `/multimedia/music` may be mounted by host "dbox", but not with write access.

After changing `/etc/exports` the NFS-daemon does not automatically take changes into account. The system is told to re-read `/etc/exports` by, e.g., a command like `exportfs -a` or `/etc/init.d/nfsserver restart`.

Solaris does not use `/etc/exports` but a `share-command`.

If using `newmake` to compile the CDK, it generates an `/etc/exports-fragment`, suitable for exporting the used `cdkroot` to the `dbox` -- assuming that the compile host is the same as the NFS-Server of course.

5.1. Troubleshooting

The NFS server is in general quite robust, and seldomly makes problems. Most problems are permissions problems. Look in the log file(s) on the server (typically `/var/log/messages`) for problems. However, the Tuxbox does not implement file locking over NFS, so file systems have to be mounted using the "nolock" option (on the client, that is), otherwise the mount may hang.

6. Other possible services

There is a vast number of different services that may or may not be of interest. Multimedia related services are, e.g. shoutcast and VLC. There are also the traditional Unix-Services. These can, possibly after installing some client software component, in general be used with the Tuxbox.

6.1. DNS Name server

To have your own nameserver in your LAN may be useful, or at least nice. That way, it is not necessary to keep `/etc/hosts` up to date on all hosts on the LAN.

6.2. RARP server, Bootp server

The `dhcpd` daemon, setup like here proposed, takes over the task of these services, which are therefore not needed.

7. Flashing

In [this article](#) a non-interactive flashing using the "dboxflasher", a specially compiled u-boot, is described. This is an application using DHCP and TFTP.

Booting from YADD and flashing, using either Neutrino's (or Enigma's) flashing functions (or, from the command line, the commands `eraseall` or `fcg`) is a viable possibility both for reading and writing of the flash memory (partition `/dev/mtd/4`, "Flash without bootloader").

8. Troubleshooting

For trouble shooting, a log from the serial console (also called bootlog) is indispensable. See [wiki](#) for some more details. Without it, no sensible debugging is possible, just guessings. Also, the log on the server (typically `/var/log/messages`) can give many important hints, in particular regarding NFS- and DHCP-problems.

9. Links

- [CDK YADD Boot Procedure">](#) in Tuxbox Wiki. Also in German as [CDK \(YADD\) Bootvorgang](#). Describes the steps involved in booting a CDK setup.
- [DHCP software](#) from ISC.