

PXE-Installation für EISFAIR

Thomas Klose

Revision 0.1

Vorwort

Begonnen habe ich dieses Projekt, als ich mich entschlossen habe, einen EISFAIR-Server als Homeserver aufzusetzen. Als Hardwarebasis habe ich einen Mini-ITX-Rechner gewählt und mir die Komponenten zusammengestellt. Eine solcher Rechner ist sehr klein und enthält lediglich Platz für das Motherboard und die Festplatte. Damit ist klar, dass der Boot nur über ein temporär angeschlossenes Laufwerk erfolgen kann.

Als ich feststellte, dass das BIOS als Option LAN-Boot mittels PXE unterstützt, war das für mich die Wahl. Leider unterstützt das neuste BIOS für mein Board (EPIA 800) keine CD-Boot von einem USB-Laufwerk wohl aber einer USB-Floppylaufwerk. Nach einigen Experimenten war klar, dass sowohl der Boot mit einer USB-Floppylaufwerk wie auch einer USB-CDROM mit den Standardmethoden von EISFAIR nicht funktioniert. Nähere Beschäftigung mit Linux und dem El-Torito-Standard für CD-Boot ergaben sehr schnell, dass hier Handarbeit notwendig wird.

In den Newsgroups habe ich zwar einige Anfragen zu dem Themen „Booten von USB-Laufwerken“ und „PXE-Boot“ gefunden, aber wenig erhellendes. Dies ist keine Kritik an den Machern des EISFAIR, aber ich habe mir gedacht, wenn ich mich schon damit beschäftigt habe, denn kann ich es auch in einem How-To veröffentlichen. Wenn es dem EISFAIR-Team gefällt, kann es entsprechend veröffentlicht werden und ich würde mich freuen, wenn die Anregungen zur „Universalisierung“ des Installationsvorgangs aufgenommen würden, um die PXE- oder USB-Installation zu unterstützen.

In diesem Dokument beginne ich mit einem Kochrezept, mit Hilfe dessen jedermann (und jedefrau, was ich im folgenden nicht mehr erwähne) die notwendigen Schritte nachvollziehen kann. Die dafür notwendigen von mir speziell erstellten Dateien habe ich in einem .zip-Archiv zusammengefasst und stelle sie getrennt zur Verfügung. Für die Tools sind Download-Adressen angegeben.

Inhaltsverzeichnis

Vorwort.....	2
Inhaltsverzeichnis.....	3
1 Kochrezept.....	4
1.1 Ausgangslage.....	4
1.2 Tools.....	4
1.2.1 Zielsystem.....	4
1.2.2 TFTP-Server und DHCP.....	4
1.2.3 EISFAIR-Distribution.....	6
1.2.4 WinVi.....	6
1.2.5 PXE Zubehör-Archiv.....	6
1.3 Start des Zielsystems.....	7
1.3.1 Boot.....	7
1.3.2 Inspect.....	8
2 Hintergrundinformationen.....	9
2.1 Installation.....	9
2.1.1 Bootvorgang.....	9
2.1.2 Installationsvorgang.....	9
2.2 Aufbau der CD-ROM.....	10
2.2.1 El-Torito-Standard.....	10
2.2.2 USB-Laufwerke.....	10
3 Aufbau des PXE-Zubehör-Archivs.....	11
3.1 Bootverlauf.....	11
3.2 Bootloader und Kernel.....	11
3.2.1 Kernel.....	11
3.2.2 pxelinux.0.....	11
3.2.3 default.....	12
3.3 Busybox.....	12
3.4 Skripte.....	13
3.4.1 linuxrc.....	13
3.4.2 linuxrc.pxe.....	13
3.4.3 linuxrc1.pxe.....	14
3.4.4 linuxrc2.pxe.....	14
3.5 Initrd.gz.....	15
3.5.1 linuxrc/init.....	15
3.5.2 Netzwerktreiber.....	15
3.5.3 USB-Treiber.....	15
Abbildungsverzeichnis.....	16

1 Kochrezept

1.1 Ausgangslage

Diese Beschreibung beginnt bei einem Zielsystem für den aufzusetzenden EISFAIR-Server und einem Windows-XP Arbeitsplatzrechner. Von letzterem aus wird die Installation gesteuert und die Tools beschafft. Natürlich kann hier auch jedes andere System benutzt werden, die Hauptsache ist, es gibt Entsprechungen zu den verwendeten Tools.

1.2 Tools

Um die Arbeiten zu beginnen werden folgende Dinge benötigt:

1. Das Zielsystem.
2. Einen TFTP-Server mit DHCP-Optionen.
3. Die EISFAIR-Distribution bestehend aus CD-Image und Dokumentation.
4. Der Editor WinVi. (nur wenn besondere Änderungen erforderlich sind).
5. Das PXE-Zubehör-Archiv.

1.2.1 Zielsystem

Auf dem Zielsystem wird der EISFAIR-Server installiert. Insofern unterliegt es den entsprechenden Vorschriften eines EISFAIR-Servers. Obwohl für einen Server streng genommen nicht notwendig, wird die Installation mittels VGA-Anschluss und Tastatur vorgenommen. Es ist zwar u.U. möglich, nach den unbedingt erforderlichen BIOS-Einstellungen für den weiteren Installationsvorgang mit einer seriellen Konsole auszukommen, aber dann sind erhebliche weitere Änderungen am Standardsystem notwendig, die auf eine neue Zusammenstellung hinauslaufen.

Neben den üblichen Einstellungen für die verwendeten Peripheriegeräte ist natürlich der LAN-Boot (auch als PXE-Boot bezeichnet) zu aktivieren. Dieser sollte an erster Stelle in der Bootreihenfolge stehen, um eventuelle Reste auf der Festplatte zu umgehen. Der Installationsvorgang von EISFAIR richtet die erste Platte vollständig neu ein.

Das Zielsystem ist mit dem Netzwerksegment zu verbinden, in dem auch die (Windows XP) Arbeitsstation steht, von der aus die Installation vorgenommen wird.

1.2.2 TFTP-Server und DHCP

Für den PXE-Bootvorgang wird ein TFTP-Server für die benötigten Dateien und ein DHCP-Server für die Information des PXE-Systems benötigt. Beides ist in dem Produkt „TFTP32“ von Philippe Jounin enthalten. Es ist Freeware. Zu beziehen ist es auf der Downloadseite des Verfassers (<http://tftpd32.jounin.net>). Diese Beschreibung basiert auf der zum Zeitpunkt der Verfassung aktuellen Version v3.03. Die Zip-Datei wird in ein separates Verzeichnis (vorzugsweise C:\Programme\TFTP32) entpackt. Der Start gelingt nur als Administrator. Gute Erfahrungen liegen mit der Windows-XP-Funktion „Ausführen als“ vor, so dass die anderen Arbeiten nicht unbedingt von einem Administratoren-Konto aus erledigt werden müssen.

Für den Server ist ein Arbeitsverzeichnis anzulegen, welches für alle später nachzuladenden Dateien notwendig wird. Dieses ist auch als Wurzelverzeichnis für alle Downloads zu konfigurieren, da TFTP über keinerlei Zugriffsschutz verfügt. In diesem Arbeitsverzeichnis sind die Verzeichnisse „pxelinux.cfg“, „eis“ und darunter „eis\CD“ anzulegen.

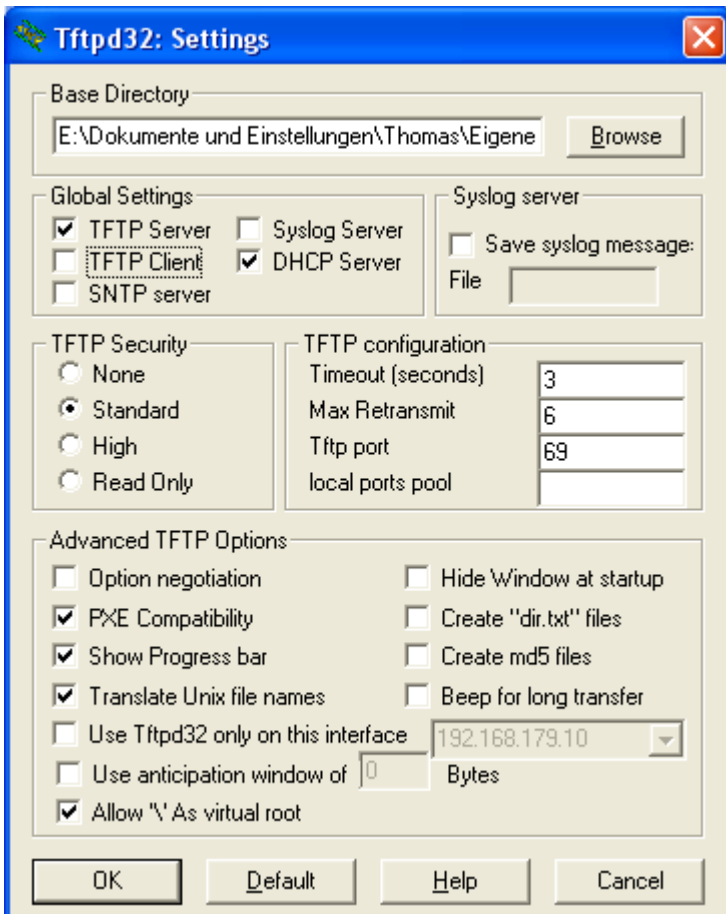


Abbildung 1: TFTP32 Settings

Als wichtigste Einstellungen sind zu nennen:

1. Base-Directory
Hier wird das zuvor angesprochene Arbeitsverzeichnis angegeben. Alle Zugriffe von TFTP könne sich nur auf dieses Verzeichnis und alle Unterverzeichnisse beziehen. Andere Dateien als die für den PXE-Vorgang benötigten sind hier abzulegen.
2. Global Settings
Hier sind mindestens der TFTP-Server und der DHCP-Server anzugeben. Der Syslog-Server wird nicht benötigt. Der TFTP-Client ist lediglich für Testzwecke erforderlich.
3. TFTP Security und Configuration
Hier könne die Standardwerte bleiben.
4. Advanced TFTP Options
Anzuwählen sind die Punkte „PXE Compatibility“, „Translate Unix file names“ und „Allow \\' as virtual root“. Letztere Einstellung dient dem Schutz. Bei Fehlfunktionen im Bootvorgang kann auch die Option „Option negotiation“ versucht werden.

In dem Tab „DHCP-Server“ sind die Einstellungen für diesen zu finden. Hier ist Vorsicht geboten. In typischen Heimnetzwerken, in denen ein Server zum Einsatz kommt verrichtet ein Router zum Internet hin -meistens ein dsl-Anschluss- seinen Dienst. Hier ist oftmals auch ein DHCP-Server eingerichtet. Daher muss der hier vorgegebene Adressbereich von dem des Standard-Routers unterschiedlich sein. Auch sollte zur Zeit des Betriebs des TFTP-Server kein anderer Client Adressen beantragen. Auf Seiten des PXE-Rechners ist dies unkritisch, da der Standard-Router nicht die für PXE-Boot erforderlichen DHCP-Optionen anbietet und daher ignoriert wird.

Tftp Server	DHCP server	Log viewer
IP pool starting address	192.168.179.210	S a v e
Size of pool	5	
Boot File	pxelinux.0	
WINS/DNS Server	0.0.0.0	
Default router	192.168.179.1	
Mask	255.255.255.0	
Domain Name		
Additional Option	0	

Abbildung 2: DHCP-Einstellungen

Die Einstellungen sind wie folgt vorzunehmen:

1. IP pool
Der IP-Pool muss wie bereits erwähnt außerhalb anderer DHCP-Ranges liegen und kann ziemlich klein sein.
2. Boot File
Hier wird angegeben, nach welcher Datei der Bootloader fragen soll. Diese muss selbstverständlich auch vorhanden sein. Zur Ablage der Datei kommen wir später.

Die weiteren Angaben spielen hier eigentlich keine Rolle. Da während des Bootvorgangs kein Zugriff aus dem lokalen Netz heraus notwendig ist, werden die weiteren Informationen nicht benutzt.

1.2.3 EISFAIR-Distribution

Zur EISFAIR-Installation gehört ein CD-Image aus der eine Installations-CD gebrannt werden muss. Aus dieser CD werden die drei Dateien „boot.tgz“, „opt.tgz“ und „rootfs.tgz“ in das Unterverzeichnis „eis/CD“ des Arbeitsverzeichnisses von oben kopiert. Wenn es das Brennprogramm zulässt, das Image direkt zu lesen, kann der Umweg über die gebrannte CD eingespart werden. Hier wird die Version von Mitte Februar 2007 (eisfair-1.3.2-cd-image.iso.zip) verwandt. Sie ist erhältlich auf der Web-Präsenz des Projektes (<http://www.eisfair.org>).

1.2.4 WinVi

WinVi ist ein Freeware-Editor von Raphael Molle, der dem Unix-Editor „vi“ nachempfunden ist. Sein großer Vorteil ist, dass er Textdateien in der Unix-Form (nur ein LF als Zeilentrenner) automatisch erkennt und dann auch genauso abspeichert. Werden also beispielsweise Shell-Skripte unter Windows mit diesem Editor bearbeitet, so sind sie ohne weitere Anpassungen sofort unter Unix/Linux lauffähig. Der Editor ist erhältlich auf seine eigenen Seite (<http://www.winvi.de>). Alternativ müsste auch WordPad von Windows funktionieren, aber hierzu wurden keine weiteren Versuche unternommen.

1.2.5 PXE Zubehör-Archiv

Das Archive PXEArchiv.zip ist inklusive Pfadnamen in das bereits angelegte Arbeitsverzeichnis des TFTP-Servers zu entpacken. Dabei ist die Option „Umwandlung LF zu CR/LF..“ sicherheitshalber abzuschalten. Danach ist die Datei „<Arbeitsverzeichnis>/pxelinux.cfg/default“ anzupassen. In der Zeile:

```
APPEND load_ramdisk=1 initrd=eis/initrd.gz root=/dev/ram0 BootType=pxe
Driver=via-rhine NAddr=192.168.179.6 NMask=255.255.255.0 Host=192.168.179.10
Next=eis/linuxrc1.pxe
```

sind die fett gedruckten Teile anzupassen. Dabei ist einzutragen:

1. Driver= **via-rhine**

Hier wird der Treiber für die Netzwerkkarte angegeben. Die Notation entspricht dem Abschnitt über die Netzwerkkarten bei der Disketten-Installation in der Dokumentation von EISFAIR (Kapitel 2.2.2). Auf dem hier vorliegenden EPIA-Board ist ein Chip von VIA verbaut, der den angegebenen VIA-Treiber benötigt.

Sollten über den Netzwerkadapter keine Informationen vorliegen, so können diese Teile auch später korrigiert werden. Hierauf wird an späterer Stelle eingegangen.

2. NAddr=**192.168.179.6** Nmask=**255.255.255.0**

Die hier angegebene Netzwerkadresse ist die, die das Zielsystem für den weiteren Installationsvorgang benötigt. Sie kann die gleiche, wie vom DHCP-Server vergebene sein, was aber zu Konflikten bei Anmeldung weiterer Clients führen kann. Wird das Zielsystem in dem gleichen Netzwerksegment installiert, in dem es später eingesetzt wird, so kann hier die endgültige Adresse verwandt werden. Nach dem Installationsvorgang verliert der hier angegebene Wert seine Bedeutung.

3. Host=**192.168.179.10**

Die Host-Adresse ist die IP-Adresse des Systems, auf dem der TFTP-Server läuft. Von diesem werden während des Installationsvorgangs weitere Dateien nachgeladen.

1.3 Start des Zielsystems

Nachdem die Vorbereitungen getroffen sind, kann ein erster Test erfolgen. Es ist sicherzustellen, dass die Netzwerkverbindungen hergestellt sind und der TFTP-Server gestartet ist.

1.3.1 Boot

Dann wird das Zielsystem eingeschaltet und im BIOS der PXE-Boot aktiviert. Im Log Viewer des TFTP-Servers ist zu erkennen, wie zunächst die Datei „pxelinux.0“ und nach einigen Fehlversuchen mit anderen Dateien die Datei „pxelinux.cfg/default“ geladen wird.

An dieser Stelle erscheint ein Bootprompt, an dem man drei unterschiedliche Eingaben machen kann:

1. local

Mit dieser Option wird der Netzwerkboot abgebrochen und von den lokalen Laufwerken gebootet. Sollte auf der üblicherweise verbauten Festplatte noch nichts installiert sein, wird diese Option nur eine entsprechende Fehlermeldung hervorrufen.

2. inspect

Fehlen insbesondere zum Netzwerkadapter noch Informationen, so kann hier eine Shell gestartet werden. Aus ihr hat man Zugriff auf eine Busybox mit einer Reihe von eingebauten Befehlen, die über die aus Platzgründen wesentlich kleinere der Originaldistribution deutlich hinausgehen. Hierzu an späterer Stelle mehr.

3. eisfair

Dies ist die eigentliche Installation. Wenn die oben angegebenen Anpassungen an der Konfiguration durchgeführt wurden, sollte das System starten und mit der Installation beginnen. Am Ende der Installation ist das Zielsystem neu zu booten. Damit dann der Netzwerkboot nicht mehr stattfindet ist entweder am Zielsystem der Netzwerkboot zu deaktivieren oder am Bootprompt wird die oben beschriebene Option „local“ verwendet.

1.3.2 Inspect

Mit Hilfe der Bootoption „inspect“ wird das System gebootet und mit einer Shell (ash) gestartet. Zu diesem Zeitpunkt existieren zwei Kommandos: ash (Almquist-Shell) und busybox, eine Sammlung gängiger Kommandos.

Da zunächst nicht einmal ein Pfad definiert ist, sind die ersten Schritte etwas ungewohnt. Zudem ist kein deutscher Tastatortreiber installiert. Am besten mountet man zunächst das Proc-Dateisystem mit dem Kommando:

```
/busybox mount -t proc /proc
```

Ab diesem Zeitpunkt sind die in der Busybox enthaltenen Befehle von der Shell aus durch einfache Eingabe ihres Namens möglich. Durch das Kommando

```
help
```

erhält man eine Liste der eingebauten Befehle. Nun kann mit Hilfe der normalen Linux-Befehle das System untersucht und wenn gewünscht manuell gestartet werden. Informationen zu den erkannten Geräten am PCI-Bus erhält man mit dem Kommando:

```
cat /proc/pci
```

Auf diese Weise sollte wenigstens der Netzwerkkarte zu ermitteln sein, um das weitere Vorgehen wie in dieser Anleitung beschrieben starten zu können.

2 Hintergrundinformationen

Diese Informationen beschreiben den Ablauf des Installationsvorgangs beim Start der normalen EISFAIR-Installation von einem IDE-CDROM-Laufwerk.

2.1 Installation

Bei der Verfolgung des Installationsvorgangs ist stets zu beachten, dass man zwei Systeme auseinander hält: das Booten des Installationssystems mit dem automatischen Ablauf der Installation, wobei einige Abfragen beantwortet werden müssen, und das eigentliche EISFAIR-System, welches durch die Installation auf die Festplatte gebracht wird. Letzteres bezeichne ich im weiteren auch als Produktionssystem (weil es später produktiv ist). Beides sind beim EISFAIR Linux-Systeme, die sich aber in einer wichtigen Eigenschaft unterscheiden. Während das Produktionssystem sehr umfangreich sein kann, ist das Installationssystem auf Platzeinsparung optimiert. Dies liegt daran, dass das Installationssystem auf eine Diskette passen soll, um „in einem Rutsch“ geladen zu werden. Aus dieser Sicht ist das Produktionssystem nur Daten, die aus einer Festplattenpartition aufgespielt werden. Da aber beides Linux-Systeme sind, fällt es dem Installationsprogramm natürlich leicht, einige Anpassungen direkt vorzunehmen.

Diese Trennung und die Beschränkung hat Auswirkungen auf die Möglichkeiten des Installationssystems: es hat nur Zugriff auf IDE- und SCSI-Geräte und kennt weder Netzwerk noch USB.

2.1.1 Bootvorgang

Obwohl es sich auf den ersten Blick unlogisch anhört, findet der Bootvorgang wie von einer Diskette statt. Es ist eine der möglichen Eigenschaften des El-Torito-Standards (dies ist die Beschreibung des CD-ROM-Boot, den die üblichen BIOSes verstehen), auf der CD ein Disketten-Image zu haben. Genau diese Variante wird bei der Original-CD von EISFAIR genutzt.

Diese virtuelle Floppydisk enthält den Kernel des Installationssystems (Linux 2.4.26-1) und ein Image einer als Root-Filesystem zu mountenden RAM-Disk in komprimierter Form („initrd.gz“). Nach dem Laden des Kernels startet dieser die Datei „/sbin/int“, welche ein Shellskript ist. Diese Datei ist auch hart verlinkt mit der Datei /linuxrc.

Das Skript lädt zunächst auf einer zweiten virtuellen Konsole (erreichbar mit Shift-F2) eine Shell für manuelle Inspektionen. Sodann versucht es CD-ROM-Laufwerke an den IDE-Schnittstellen zu finden. Sodann wird von diesem CD-ROM-Laufwerk das Archiv „opt.tgz“ nachgeladen. Dieses Archiv kann bereits im Root-Verzeichnis der CD direkt eingesehen werden. Es enthält eine Fülle weiterer Dateien, um das Betriebssystem mit weiteren Befehlen auszustatten. Zu den geladenen Dateien gehört auch die Fortsetzung mit dem eigentlichen Bootvorgang mit dem Skript „/linuxrc2“.

2.1.2 Installationsvorgang

Von dem Shellskript „/linuxrc2“ wird die eigentliche Installation durchgeführt. Zuerst wird die Partitionierung der Installationsplatte zusammengestellt und ggf. entsprechend eingerichtet. Aus dem Archiv „rootfs.tgz“ (30 MByte) wird das spätere Produktionssystem und, aus dem Archiv „boot.tgz“ die Bootpartition initialisiert. Die Bootpartition wird mit einigen Zusatzinformationen bestückt und mit dem Bootloader LILO bootfähig gemacht.

Zum Abschluss werden die Mount-Informationen in die Datei „/etc/fstab“ der späteren Root-Partition eingetragen sowie die Passwörter für die diversen User abgefragt und in die Rootpartition eingebaut.

Mit diesen Maßnahmen ist der Installationsvorgang abgeschlossen und das neue System bootfähig. Damit ist auch die Lebenszeit der RAM-Disk des Installationssystems beendet und alle seine Eigenschaften treten danach nicht mehr in Erscheinung.

2.2 Aufbau der CD-ROM

Für eine Installations-CD-ROM wird von EISFAIR ein ISO-Image herausgegeben. Dieses Image kann mit einem geeigneten Brennprogramm auf eine CD-ROM gebracht werden, von der nachher das Zielsystem gebootet wird.

2.2.1 El-Torito-Standard

Damit diese CD-ROM bootfähig ist, muss sie entsprechend dem El-Torito-Standard aufgebaut sein. Näheres kann dem entsprechendem Dokument (<http://www.phoenix.com/NR/rdonlyres/98D3219C-9CC9-4DF5-B496-A286D893E36A/0/specscdrom.pdf>) entnommen werden. Im folgendem wird auf die dort enthaltenen Formulierungen Bezug genommen.

Für die nachfolgende Untersuchungen eignet sich der bereits erwähnte Editor WinVi, in den das Image geladen und im Hexadezimal-Modus angezeigt wird. Hierzu ist es notwendig unter dem Menü „Optionen“ den Punkt „Hexadezimalmodus“ und unter „Optionen/Einstellungen/Anzeige“ den Punkt „hexadezimale Adressen ...“ auszuwählen.

Im Sektor 0x17 (Sektorlänge 2048 Byte = Adresse 0x8800) findet sich Volume-Descriptor des Booting Catalog. Dieser (Sektor 0x14 = 0xA000) enthält als einzigen Eintrag einen Verweis auf ein Image einer 1,44 MByte Floppydisk in Sektor 0x15 (Adresse 0xA800).

Das so aufgefundene Image kann bei Bedarf durch Copy and Paste extrahiert und anschließend z.B. mit Hilfe des Programms „EXTRACT“ von Gilles Vollant (<http://ourworld.compuserve.com/homepages/gvollant/extrac21.zip>), welches auch beim Schwesterprojekt Fli4l Anwendung findet, ausgepackt werden.

Für das Verständnis des Bootvorgangs ist wichtig zu wissen, dass ein BIOS, welches von CD-ROM-Laufwerken booten kann, zum Zeitpunkt des Aufrufs des Bootsektors BIOS-Befehle für den Zugriff auf die so gebildete virtuelle Floppydisk zur Verfügung stellt.

2.2.2 USB-Laufwerke

An diese Stelle wird auch klar, wieso ein Boot von USB-Laufwerken scheitert. Der Bootloader SYSLINUX der virtuellen Bootfloppy kann noch auf das BIOS zugreifen und daher den Loader „ldlinux.sys“, die Konfigurationsdatei „syslinux.cfg“ sowie Kernel und Image der RAM-Disk auslesen. In dem Moment, wo der Kernel die Bearbeitung übernimmt, wird nicht mehr das BIOS benutzt. Da in dem Kernel aber keine Treiber für USB-Laufwerke eingebaut sind und die initiale RAM-Disk keine Treiber hierfür mitbekommen hat, scheitern sowohl das Nachladen von einem USB-Floppydisk-Laufwerk als auch der Zugriff auf das USB-CD-ROM-Laufwerk, obwohl von beiden erfolgreich der Kernel geladen wurde.

3 Aufbau des PXE-Zubehör-Archivs

Das PXE-Zubehör-Archiv enthält den Kernel, ein neues Image für die Ramdisk beim Startvorgang, einige Skripte und den Bootloader. Um daraus eine bootfähige Installation zu erhalten, müssen nur noch einige Original-Dateien, wie in Abschnitt 1.2.3 beschrieben, hinzugefügt werden.

Die wesentlichen Dateien wurden aus dem Boot-Floppy-Image der CD extrahiert und modifiziert.

3.1 Bootverlauf

Um den Sinn der einzelnen Dateien und Modifikationen zu verstehen, ist ein Überblick über die Vorgänge beim Bootverlauf notwendig. Der grundsätzliche Bootverlauf der PXE-Installation ist wie folgt:

1. Beim PXE-Boot fragt der bootende PC via DHCP nach einer IP-Adresse. Ein unterstützender DHCP-Server bietet auch gleichzeitig PXE-Optionen an. Eine dieser Optionen ist der Boot-Server der Filename. Damit kann der bootende Rechner Angebote von normalen Homeroutern ignorieren, da diese nicht die erforderlichen Optionen für PXE-Boot anbieten. Mit der in Deutschland weit verbreiteten Fritz-Box von AVM gibt es keine Probleme mit der Koexistenz in einem Netz.
2. Der bootende Rechner handelt mit dem DHCP-Server seine IP-Adresse aus. Diese Adresse ist nur während des Installationsvorgangs relevant.
3. Der bootende Rechner lädt die angegebene Datei via TFTP und führt sie aus. In diesem Fall hier ist es die Datei pxelinux.0 aus dem Paket SYSLINUX. Diese wiederum erfragt den Kernel und zusätzliche Parameter wie das Image der initialen RAM-Disk und die weiteren Bootparameter, die als Shellvariablen dem zuerst laufenden Skript (/sbin/init = /linuxrc) übergeben werden.
4. Der Bootloader lädt den Kernel und die RAM-Disk in den Speicher und übergibt die Kontrolle an den soeben geladenen Kernel. Dies sucht zunächst nach dem Init-Prozess unter /sbin/init. Dies ist hier ein Shell-Skript, welches die weitere Kontrolle übernimmt.
5. Das Shellskript initialisiert die Netzwerkkarte und den TCP/IP-Stack.
6. Beim PXE-Install lädt das Shellskript die weiteren Dateien nicht wie im Original von einer CD-ROM sondern via TFTP vom Server.

3.2 Bootloader und Kernel

Der Bootloader für die PXE-Installation ist wie im Original SYSLINUX von H. Peter Anvin (<http://syslinux.zytor.com>). Hier liegt die Version 3.35 zugrunde.

3.2.1 Kernel

Der Kernel stammt von der Original-CD-ROM. Auf dieser befindet sich gem. El-Torito-Standard das Image einer 1,44 MByte Floppydisk, welches u.a. den Kernel enthält. Dieser ist unverändert übernommen.

3.2.2 pxelinux.0

Aus dem SYSLINUX-Paket wird lediglich der PXE-Bootloader (pxelinux.0) benötigt. Dieser fungiert als eigentlicher Bootloader, der über TFTP den Kernel und die RAM-Disk nachlädt. Zur Auswahl desselben zeigt er ein Bootprompt, an welchem unterschiedliche Optionen ladbar sind.

Wie im SYSLINUX der Original-CD wird hierzu eine syslinux.cfg benötigt. Diese befindet sich aber an anderer Position.

3.2.3 default

Die zuvor erwähnte syslinux.cfg befindet sich in der Datei pxelinux.cfg/default. der Bootloader pxelinux.0 sucht die zu wählende Konfigurationsdatei zunächst anhand der MAC-Adresse der Netzwerkkarte des bootenden Systems, dann anhand der vergebenen IP-Adresse und zuletzt die Datei default. Näheres ist in der Dokumentation zu SYSLINUX zu entnehmen.

```
# EPIA
prompt 1
```

Die Option „prompt 1“ erzwingt eine Boot-Prompt. Im folgenden werden drei Option angeboten.

```
label inspect
  kernel eis/kernel
  APPEND load_ramdisk=1 initrd=eis/initrd.gz root=/dev/ram0 →
  →BootType=xxx
```

Diese Variante setzt lediglich eine Shell-Variable (BootType), die vom Init-Skript ausgewertet wird.

```
label eisfair
  kernel eis/kernel
  APPEND load_ramdisk=1 initrd=eis/initrd.gz root=/dev/ram0 →
  →BootType=pxe Driver=via-rhine Naddr=192.168.179.6 →
  →NMask=255.255.255.0 Host=192.168.179.10 Next=eis/linuxrc1.pxe
```

Die eigentliche Installation macht Angaben zur Netzwerkkarte (via-rhine), Netzwerkadresse und -maske, dem Host, der die Dateien enthält (im vorliegenden Fall derselbe, auf dem der TFTP-Server läuft) und das Folgeskript. Die fett gedruckten Angaben sind auf die jeweilige Situation anzupassen.

```
label local
  localboot 0
```

Als letztes gibt es noch die Möglichkeit direkt von den eingebauten Platten zu booten.

3.3 Busybox

Die üblicherweise in Linux vorzufindenden Befehle sind wie beim Original in einer Busybox zusammengefasst. Die hier in das Image für die RAM-Disk eingebaute Version ist deutlich umfangreicher als das Original. Da beim Netzwerkboot die Platzbeschränkung von 1,44 MByte eines Floppydisk-Images nicht gegeben ist, wurde hier auf Unterstützung bei der Inspektion des Systems Wert gelegt. Besonders wichtig ist der Befehl „tftp“, der zum Nachladen der Skripte und weiteren Dateien benötigt wird.

Eine weitere Verbesserung ist die Unterstützung der eingebauten Befehle. Nach dem Mounten des /proc-Dateisystems kann die eingebaute ash die Befehl direkt aufrufen (z.B. mount statt /busybox mount).

Die hier vorliegende Busybox (<http://www.busybox.net>) ist die Version 1.4.1 mit dem tftp-Patch vom 03.03.2007 und einer kleinen Modifikation im Quelltext von tftp.c (ersetzte xhost2sockaddr durch host2sockaddr). Sie wurde in einer chroot-Umgebung statisch gegen eine UCLIBC (Version 0.9.27) gelinkt (http://www.kernel.org/pub/linux/libs/uclibc/root_fs_i386.ext2.gz).

3.4 Skripte

Neben der oben beschriebenen neuen Busybox ist das Init-Skript überarbeitet worden. Es ist nunmehr in mehrere Teile aufgespalten.

3.4.1 linuxrc

Das Skript linuxrc, welches auch verlinkt ist mit /sbin/init ist nunmehr lediglich ein Multiplexer, der verschiedene Bootvarianten starten kann.

```
#-----  
# boot multiplexer  
#-----  
case $BootType in  
  pxe)  
    exec /ash /linuxrc.pxe  
    ;;  
  ide-cd)  
    exec /ash /linuxrc.ide-cd  
    ;;  
  *)  
    exec /ash  
    ;;  
esac
```

Die Shellvariable „BootType“ wird in der SYSLINUX-Konfiguration in der Append-Zeile gesetzt. Durch Angabe des Wertes „pxe“ wird der hier beabsichtigte Netzwerk-Installationsvorgang gestartet. Hierzu wird ein weiteres Skript aufgerufen. Wird als BootType „ide-cd“ angegeben, so wird das Skript linuxrc.ide-cd aufgerufen, welches das Original-Skript aus der EISFAIR-CD ist. Hat man beispielsweise ein BIOS, welches nicht von CD-ROM booten kann, so kann durch die Angabe der Option „ide-cd“ das Nachladen dennoch von CD erfolgen.

3.4.2 linuxrc.pxe

Wurde die Option „pxe“ als BootType definiert, so wird das Skript linuxrc.pxe ausgeführt.

```
#-----  
# prepare filesystems (proc and /)  
#-----  
/busybox mount -av 2>/dev/null  
/busybox mount -o remount,rw /
```

Zunächst wird das /proc-Dateisystem gemountet und das Root-Filesystem beschreibbar gemacht.

```
#-----  
# seting up network interface  
#-----  
  
/busybox insmod pci-scan  
/busybox insmod $Driver  
  
/busybox ifconfig eth0 $NAddr netmask $Nmask
```

Das Laden der Netzwerktreiber erfolgt aus den Angaben in SYSLINUX-Konfiguration.

```
#-----  
# loading follow up skript and executing  
#-----
```

```
/busybox tftp -l linuxrc1.pxe -r $Next -g $Host
exec /ash linuxrc1.pxe
```

Als letztes wird das erste Skript vom TFTP-Server geholt und zur Ausführung gebracht. Damit kann bereits dieser erste Teil modifiziert werden, ohne jedes mal ein neues initrd packen zu müssen.

3.4.3 linuxrc1.pxe

Diese erste vom Server geladene Skript ähnelt bereits sehr dem Original-Skript von der CD; es ist im wesentlichen dessen letzter Teil.

```
#-----
# load additional files
#-----

optfile=opt.tgz

tftp -l - -r eis/CD/$optfile -g $Host | zcat | tar xf -
```

Das opt.tgz (von der Original-CD) enthält die wesentliche Ergänzungen zum Root-Filesystem der Installation. Statt wie beim Original von der CD wird es vom TFTP-Server geladen

```
if [ "$?" != "0" ]
then
    echo "*** ERROR: $optfile not found or RAMSIZE to small ***"
    while [ 1 = 1 ]
    do
        read dummy
    done
    exit
fi
```

Die Fehlerbehandlung erfolgt wie beim Original.

```
tftp -l linuxrc2.pxe -r eis/linuxrc2.pxe -g $Host

exec /bin/sh /linuxrc2.pxe
```

In dem opt.tgz ist auch das eigentliche Installationsskript enthalten. Da diese aber für die Netzwerkinstallation nicht geeignet ist, da es die weiteren Dateien von der CD nachlädt, wird es mit der angepassten Version überschrieben.

3.4.4 linuxrc2.pxe

Das Installationsskript, welches die eigentliche Arbeit übernimmt musste nur an zwei Stellen angepasst werden. Dort, wo Dateien von der CD-ROM geladen wurden, ist eine Anpassung erfolgt.

```
#-----
# install root filesystem
#-----

colecho "installing root filesystem ..." gn
echo
mkdir /rootfs /boot
mount /dev/$disk$root_idx /rootfs
cd /rootfs
```

```
mkdir tmp proc mnt cdrom floppy usb boot data
chmod 1777 tmp
/busybox tftp -l - -r eis/CD/rootfs.tgz -g $Host | tar xpz
```

Das Root-Filesystem der Produktions-Partition wird aus dem Netzwerk nachgeladen. Es bewegt sich mit seinen 30 MByte nahe am Maximum eines TFTP-Downloads, was aber mit einer höheren Blocksize noch verbessert werden kann.

```
#-----
# install lilo bootloader
#-----

(...)

cd $bootdir
/busybox tftp -l - -r eis/CD/boot.tgz -g $Host | tar xpz
cd /
```

Auch das Filesystem für die Boot-Partition wird vom TFTP-Server geladen.

3.5 Initrd.gz

Um zu einem neuen Image für die initiale RAM-Disk zu kommen, wird von Original ausgegangen. Am besten mountet man das Image per Loop-Device auf einem Linux-System, nimmt die Veränderungen vor und packt das Ergebnis als neue initrd.gz ein.

3.5.1 linuxrc/init

Zunächst ist das originale Skript „linuxrc“ umzubenennen nach „linuxrc.ide-cd“. Danach ist die Datei „/sbin/init“ zu löschen. Dies nur ein Hardlink auf die zuvor umbenannte linuxrc. Sodann werden die oben beschriebenen Skripte „linuxrc“ und „linuxrc.pxe“ erzeugt und im Rootverzeichnis abgelegt. Zusätzlich ist ein Hardlink aus „linuxrc“ unter „/sbin/init“ zu erzeugen.

3.5.2 Netzwerktreiber

Die Netzwerktreiber stammen aus der rootfs.tgz der Original-CD. Es ist das gesamte Verzeichnis net der Kernel-Module (/lib/modules/2.4.26-1/kernel/drivers) ohne Unterverzeichnisse. Damit dürfte ein große Sammlung gängiger Treibern verfügbar sein.

3.5.3 USB-Treiber

Zu experimentellen Zwecken wurden noch Treiber für die Einbindung von USB-Laufwerken eingefügt. Es steht noch nicht fest, ob diese Auswahl ausreichend ist, um das Nachladen von Treibern von einem USB-Laufwerk durchzuführen. Im einzelnen wurden von der Original-CD aufgenommen:

1. scsi/scsi_mod.o, scsi/sd_mod.o, scsi/sr_mod.o
2. usb/usbcore.o
3. usb/storage/usb-storage.o
4. usb/host/uhci.o, usb/host/usb-ohci.o, usb/host/usb-uhci.o

Abbildungsverzeichnis

Abbildung 1: TFTP32 Settings.....5
Abbildung 2: DHCP-Einstellungen.....6