

Studienarbeit
Media on Demand - Diskussion verschiedener Ansätze

Bearbeitet von Steffen Stein, Matr. Nr. 2634780
Aufgabenstellung und Betreuung Zefir Kurtisi
Stand 12. November 2004

Erklärung

Ich versichere, die vorliegende Arbeit selbstständig und nur unter Benutzung der angegebenen Hilfsmittel angefertigt zu haben.

Braunschweig, den 12. November 2004

Inhaltsverzeichnis

Inhaltsverzeichnis	0
1 Einleitung	1
2 Architektur eines Media on Demand Systems	2
2.1 Frontends	2
2.2 Backends	3
3 Eignung der Player für eine Kiosk-Umgebung	5
3.1 VLC	5
3.2 MPlayer	6
3.3 Fazit	6
4 Streaming	7
4.1 Bandbreitenbedarf	7
4.2 Streamen von Dateien	8
4.3 Streamen von DVDs	9
4.4 Streamen von DVB	10
4.5 Fazit	10
5 Beispielkonfiguration einer Media on Demand Lösung	11
5.1 Grundsätzliches Setup	11
5.2 Konfiguration	11
5.3 Betrieb	13
6 Fazit und Ausblick	15
Literaturverzeichnis	16
A Installation eines Debian Systems	17
B Installation eines SuSE Linux Systems	19

1 Einleitung

Im Rahmen dieser Arbeit werden mehrere Programme für das Streamen von Videos in verschiedenen Formaten vorgestellt. Insbesondere wird auf das Bereitstellen von DVDs und kodierten Filmen in verschiedenen Formaten eingegangen. Im Folgenden wird zwischen Servern (Backends) und Playern (Clients, Frontends) unterschieden.

Die Studienarbeit richtet sich an Leser, die Grundkenntnisse in der Bedienung von Linux Systemen mitbringen. Sie soll einen Überblick über vorhandene Lösungen, um Audio- und Videodateien on Demand zur Verfügung stellen, vermitteln, sowie als Grundlage für eine Entscheidungsfindung für oder gegen eine spezifische Lösung dienen. Exemplarische Installationsanleitungen für ein rpm-basiertes und ein apt-basiertes System (SuSE Linux 9.0 bzw. GNU/Debian) runden die Arbeit ab und helfen bei der Umsetzung eines spezifischen Projektes.

Auf die Eignung der vorgestellten Programme zum Einsatz in einer Kiosk Umgebung wird genauer eingegangen, da viele Szenarien denkbar sind in denen es von Vorteil ist, die Benutzerschnittstelle als Kiosk System zu implementieren. Der Text ist in fünf Teile gegliedert. Zuerst wird beschrieben, was ein Video on Demand System leisten soll, sowie eine Architektur vorgeschlagen. In diesem Zusammenhang werden auch verschiedene Programme, die als Front- und Backends dienen können, vorgestellt. Insbesondere wird auch darauf eingegangen inwiefern die Frontends für die Einbindung in eine Kiosk Umgebung geeignet sind. Anschließend wird das Streamen von Dateien behandelt. An dieser Stelle werden die Ansprüche an das zugrunde liegende Netzwerk analysiert, sowie das Zusammenspiel der im ersten Abschnitt vorgestellten Softwarekomponenten bewertet. Im vierten Teil wird eine spezifische Media on Demand Lösung vorgestellt und ausführlich beschrieben. Hier wird auch exemplarisch auf die Konfiguration einiger vorgestellter Programme eingegangen. Eine Bewertung der Ergebnisse, sowie ein Ausblick auf künftige Entwicklungen runden die Arbeit ab. Installationsanleitungen für alle hier beschriebenen Programme unter SuSE Linux 9.0 und Debian befinden sich im Anhang. Diese beiden Linux Systeme stehen beispielhaft für eine ganze Reihe anderer Systeme, die auf den selben Paketverwaltungssystemen aufbauen. SuSE Linux ist rpm basiert (wie z.B. auch RedHat oder Fedora), Debian ist apt-basiert.

2 Architektur eines Media on Demand Systems

Unter einem Media on Demand (MoD-) System wird ein System verstanden, in dem der Benutzer jederzeit (on Demand) auf Audio- und Videoinhalte zugreifen und diese sofort abspielen kann. Damit ein derartiges System möglichst universell einsetzbar ist, sollte es eine Vielzahl von Kodierungen und Formaten (Codecs) der Audio und Video Dateien unterstützen. Insbesondere ist auch die Unterstützung für das Bereitstellen von ganzen DVDs ohne Funktionalitätsverlust (z.B. Menüführung) oder Fernsehsendern (in dieser Arbeit am Beispiel DVB-T, der digitalen Fernsehübertragung über Antenne erwähnt) interessant. Aus verschiedenen Einsatzszenarien für ein MoD-System ergeben sich unterschiedliche Anforderungen an die verwendete Software. Im folgenden wird davon ausgegangen, dass die Backends (Server) vom Administrator des Systems eingerichtet werden, hier also umfangreiche Konfigurationen vorgenommen werden können. Die Endsysteme, auf denen die Frontends des Media on Demand Systems laufen, müssen allerdings nicht zwingend vom Administrator des MoD-Systems verwaltet werden. Hier sollte der Konfigurations- und Installationsaufwand minimal gehalten werden.

Zentral administrierte Benutzeroberflächen werden zunehmend Browserbasiert implementiert. Die Oberfläche wird in HTML programmiert, von einem Webserver bereit gestellt und durch einen Standardbrowser angezeigt. Offensichtlich ist dieser Ansatz sehr gut für die Bedienoberfläche eines MoD-Systems geeignet, da auf den Clients nur ein Standardbrowser installiert sein muss. Dieser muss gegebenenfalls noch vom Benutzer richtig konfiguriert werden; Anweisungen hierfür können jedoch in die Oberfläche eingebaut werden. Des weiteren ist ein Webserver auch in der Lage die Audio und Video Inhalte des MoD-Systems, die jeweils in genau einer Datei gespeichert sind, zur Verfügung zu stellen. Alle anderen Inhalte wie DVDs und DVB-T müssen von anderen Servern bereit gestellt werden.

Aufgrund der vorangegangenen Überlegungen wird in dieser Arbeit die folgende Architektur eines Media on Demand Systems favorisiert. Ein Webserver stellt die Inhalte des MoD-Systems, die in einzelnen Dateien kodiert sind, zur Verfügung. Für das Streamen von DVDs sowie DVB-T werden spezielle Server verwendet. Als Frontends kommen auf den Clients Multimedia Player zum Einsatz. Außerdem wird in dieser Arbeit davon ausgegangen, dass, so eine Browserbasierte Benutzeroberfläche implementiert werden soll, clientseitig ein Mozilla-kompatibler Browser (z.B. Mozilla, Firefox, Opera) zum Einsatz kommt. Eine komplette, offene Video on Demand Lösung, MythTV, wird in dieser Arbeit nicht besprochen, da sie nur Unterstützung für das Streamen von Filmen in einem proprietärem Format bietet.

2.1 Frontends

Für Linux Systeme existieren zwei sehr ausgereifte Multimedia Player - den Client des VideoLAN Projektes (VLC) [1] und den MPlayer [2]. Beide Player können sowohl unter Linux als auch unter Windows eingesetzt werden und unterstützen das Abspielen sehr vieler Formate. Da die Entwickler-Teams der beiden Projekte sehr aktiv sind, ist zu erwarten, dass Funktionalitäten, die zum Zeitpunkt der Anfertigung dieser Arbeit noch nicht vorhanden waren, jetzt schon implementiert sind.

Da die Player für das Abspielen/ Decodieren der Filme weitestgehend die selben Bibliotheken verwenden, ist die Funktionalität der Player fast identisch.

In den folgenden Abschnitten werden einige Eigenarten der Player beschrieben.

VLC

Wie der Name vermuten lässt, wurde der VideoLAN Client (VLC) als Client für eine Verteilte Umgebung entwickelt. Dies zeigt sich unter anderem darin, dass er in HTTP/1.1-Streams suchen („spulen“) kann. Die Installation unter einem SuSE System hat sich als aufwendig heraus gestellt, da viele der benötigten Pakete nicht Teil der Distribution sind und einige Komponenten sogar vom Anwender selbst kompiliert werden müssen. Dasselbe gilt wahrscheinlich auch für andere rpm-basierte Systeme (RedHat, Fedora). Nach der Installation ist der VLC intuitiv bedienbar und kann viele verschiedene Formate abspielen. Insbesondere bietet er im Gegensatz zum MPlayer Unterstützung für DVD-Menüs. DVDs kann der VLC auch dann abspielen, wenn sie in ein lokales Verzeichnis auf der Festplatte kopiert wurden. Dies ist für das Streamen von DVDs interessant.

Für den VLC gibt es ein Mozilla Plugin, das zwar nur beschränkte Funktionalität bietet, sich jedoch effektiv durch Javascript Befehle steuern lässt. Die komplette Dokumentation, sowie das Programm als solches, kann man im Internet unter [1] herunterladen.

MPlayer

Der MPlayer wurde als Multimedia Player für Linux entwickelt. Er läuft stabil und unterstützt viele Formate. Die aktuelle Version (1.0rc3) bietet keine Unterstützung für DVD Menüs.¹ Der MPlayer kann jedoch, genau wie der VLC, die einzelnen .vob Dateien² einer DVD direkt abspielen, auch wenn diese in ein lokales Verzeichnis kopiert wurden.

Der MPlayer kann vollständig von der Kommandozeile bedient werden. Auch nach dem Start kann er Befehle über den Standardinput verarbeiten. Dadurch wird der MPlayer sehr flexibel und ist zur Einbettung in andere Systeme gut geeignet.

Obwohl man den Player unter SuSE Linux selber kompilieren muss, verläuft die Installation problemlos. Die Ausgaben des Konfigurationsscriptes, das die Quellen für das Kompilieren vorbereitet, sind hier sehr aussagekräftig und hilfreich.

Auch dieser Player unterstützt das Abspielen von Streams über HTTP und ist somit als Frontend geeignet. Der Quelltext des Programms sowie die komplette Dokumentation kann im Internet unter [2] heruntergeladen werden.

2.2 Backends

Obwohl im Verlauf dieser Arbeit auf den Apache Webserver, den Kernel NFS Server und den VideoLAN Server (VLS) eingegangen wird, wird in diesem Abschnitt nur der VLS vorgestellt, da eine vollständige Beschreibung der Funktionalität der ersten beiden Server den Rahmen dieser Arbeit sprengen würde. Es

¹Offizielles Statement des MPlayer Entwicklungsteam: „Support for dvdnav in MPlayer is currently broken, normal playback does work, though. If you want to have fancy menus, you will have to use another player like Xine or Ogle. If you care about DVD navigation, send a patch.“ ([2] Dokumentation Kapitel 4 FAQ)

²Diese Dateien beinhalten die mpeg-2 codierten Film Daten einer DVD.

genügt zu wissen, dass beide Dateien im Netzwerk zur Verfügung stellen können. Ausführliche Beschreibungen der Funktionalität des Apache Webservers und des Kernel NFS Servers finden sich in [3] bzw. [4].

VLS

VLS wurde als Streaming Server für das VideoLAN Projekt entwickelt. Der VLS kann alle unterstützten Formate sowohl unicast als auch multicast Streamen. Dies macht den VLS besonders interessant, wenn man Inhalte für mehrere Clients gleichzeitig zur Verfügung stellen möchte (z.B. Lehrfilme). Unter Linux kann er aktuell die folgenden Formate streamen:

- Dateien in den Formaten, die auch der VLC abspielen kann. Hierunter fallen insbesondere auch DivX, Xvid und ähnliche Formate.
- Streams aus Mpeg-2 Karten, wie DVB{C|S|T}- oder Mpeg-2-Encoder Karten (wie z.B. die Hauptauge DVR-Serie)
- DVDs sowohl von der Scheibe als auch von lokalen Kopien auf der Festplatte.

Die Dateien werden als Mpeg-2-TS (Transport Stream) gesendet. Leider kann der MPlayer den Mpeg-2-TS des VLS nicht lesen. Folglich kommt als Frontend nur der VLC in Frage, wenn man den VLS als Backend einsetzen möchte.

Nach der Standard-Installation befindet sich die Konfigurations-Datei für den VLS unter `/usr/local/etc/violan/vls`. Die Datei heißt `vls.cfg`. Bei der Standard-Installation wird eine gut dokumentierte Beispieldatei an oben genannte Position kopiert.

Eine genaue Dokumentation der Struktur der Konfigurationsdatei findet sich im Internet unter [1].

3 Eignung der Player für eine Kiosk-Umgebung

Unter einer Kiosk-Umgebung versteht man eine Bedieneroberfläche auf einem Standardbetriebssystem, die dem Benutzer nur genau die Möglichkeiten zur Interaktion bietet, die er auch nutzen soll. Solche Systeme finden inzwischen weite Verbreitung. Ein bekanntes Kiosk-System ist die Oberfläche der Buchungsmaschinen der Deutschen Bahn. Die Oberfläche läuft auf einem Windows-System und blendet jegliche Funktionalität die Windows bietet, aus.

Damit ein Programm zum Einsatz in einer Kiosk-Umgebung geeignet ist, sollte die Oberfläche des Programms sich nach den Wünschen des Administrators des Kiosk-Desktops verändern lassen. Insbesondere sollten einige Funktionen ausgeblendet werden können.

Wie in verschiedenen anderen Arbeiten und Anwendungen gezeigt, eignet sich der frei verfügbare Browser Mozilla sehr gut zur Einbettung in oder Darstellung einer Kiosk Umgebung ([5], [6]). Offensichtlich lassen sich Kiosk-Systeme relativ einfach durch HTML-Seiten realisieren, die von einem in der Funktion eingeschränkten Mozilla Browser angezeigt werden, der wiederum die gesamte Funktionalität der graphischen Oberfläche überdeckt. Deshalb wird im weiteren die Möglichkeit der Einbettung der Anwendung in eine HTML-Seite neben der Möglichkeit zur Veränderung des Aussehens der Anwendung als wesentlich für die Eignung zum Einsatz in einer Kiosk-Umgebung angesehen.

Für Mozilla-kompatible Browser (z.B. Netscape, Mozilla, Opera) gibt es unter Linux ein Plugin, das es erlaubt, beliebige externe Anwendungen zu starten und das Fenster der Anwendung in die aktuelle HTML-Seite einzubetten. Mit Hilfe dieses Plugins lassen sich alle Player, die nur ein (nicht zwei oder mehrere) Fenster öffnen, in eine Webseite einbetten. Der Name dieses Plugins ist Mozplugger und es ist bei SuSE 9.0 und Debian Bestandteil der Distribution (s.h. Installation). Es wird zusammen mit einer gut kommentierten Konfigurationsdatei (`/etc/mozplugger.conf`) installiert. Die gesamte Dokumentation, sowie eine aktuelle Version des Mozpluggers kann man im Internet unter [7] beziehen.

3.1 VLC

Für den Player des VideoLAN Projektes (VLC) wurde auch ein Mozilla Plugin für die Benutzung unter Linux entwickelt. Mit diesem Plugin lässt sich der VLC einfach und effektiv in eine Webseite einbetten. Das Plugin lässt sich im Betrieb durch Javascript Befehle steuern. Hier steht jedoch nur beschränkte Funktionalität zur Verfügung. Die unterstützten Befehle sind `play()`, `pause()`, `stop()` und `fullscreen()`. Darüber hinaus lässt sich der VLC mit denselben Tastenkürzeln wie im normalen Betrieb steuern. So wechselt z.B. „F“ in den Vollbild Modus und zurück. Die einzige Dokumentation des Plugins findet sich in einem Beispiel in der Dokumentation des VLC auf [1].

Da der VLC beim Abspielen eines Videos immer zwei Fenster öffnet (Bedienelemente, Video), lässt er sich nicht effektiv mit dem Mozplugger in eine Webseite integrieren.

Für das grafische Benutzerinterface (graphical user interface, GUI) vom VLC-Player können eigene Skins entworfen werden. Welche Funktionalitäten diese Skins bieten, bleibt dem Entwickler überlassen, so dass auch in der Funktionalität eingeschränkte GUIs erstellt werden können. Der VLC ist offensichtlich für die Integration in eine Kioskumgebung geeignet, egal ob diese HTML basiert ist oder nicht. Strebt man eine

HTML-basierte Lösung an, ist der Funktionsumfang des VLC allerdings eingeschränkt.

3.2 MPlayer

Das Erscheinungsbild sowie das Verhalten des MPlayers lässt sich sehr gut durch optionale Argumente beim Programmaufruf steuern. So muss der Player nicht zwingend die Bedienelemente einblenden. Diese werden allerdings in einem separaten Fenster dargestellt wenn man sie einblendet. Wird nur das Videofenster angezeigt, lässt sich der MPlayer effektiv mit dem Mozplugger in eine Webseite einbetten. Alternativ wäre es auch möglich, den Player als Standardplayer für Multimedia-Dateien einzurichten. Startet man ihn im Vollbild Modus, würde er alle anderen Bedienelemente überdecken, bis der Film zu Ende ist.

Im laufenden Betrieb lässt sich der MPlayer, wie der VLC, durch Tastendrucke steuern. Das Verhalten ist durch eine Konfigurationsdatei beeinflussbar. So kann auch hier die gebotene Funktionalität eingeschränkt werden. Außerdem besteht die Möglichkeit, den MPlayer durch Befehle über den Standard-Input zu bedienen. Hierbei steht die gesamte Funktionalität des Players zur Verfügung. Dies macht den MPlayer sehr flexibel für die Einbettung in beliebige Umgebungen.

Genau wie für den VLC kann man auch für den MPlayer Skins entwerfen. Dieser ist folglich auch für die Integration in eine Kiosk-Umgebung, die nicht auf HTML-Seiten basiert, geeignet.

3.3 Fazit

Durch die Möglichkeit, den MPlayer über den Standard-Input „fernzusteuern“, ist dieser etwas flexibler, wenn man die Kiosk-Umgebung auf einer HTML-Oberfläche aufbauen möchte. Ist jedoch lediglich die Funktionalität zum Abspielen von Inhalten gefragt, oder die Oberfläche nicht HTML-basiert, sind beide Player gleichermaßen zur Integration in eine Kiosk-Umgebung geeignet.

4 Streaming

Dieser Abschnitt behandelt das Streamen von Dateien, die bereits auf der Festplatte vorhanden sind. Insbesondere werden dabei Filme, die als einzelne Datei vorliegen, sowie auf die Festplatte kopierte DVDs eingegangen. Bei unverschlüsselten DVDs kann man einfach das video_ts³ Verzeichnis der DVD auf die Festplatte kopieren. Für verschlüsselte DVDs gibt es Tools, die das video_ts Verzeichnis auf die Festplatte kopieren können und dabei die .vob-Dateien entschlüsseln.⁴

Bevor das Zusammenspiel der einzelnen Front- und Backends vorgestellt wird, wird der Bandbreitenbedarf beim Übertragen verschiedener Formate untersucht, da dieser unabhängig vom verwendeten Server und Client ist. Der Bandbreitenbedarf ist der wesentlichste Faktor, der die Skalierbarkeit einer Media on Demand Lösung beeinflusst.

Bei Filmen, die als einzelne Datei vorliegen, werden insbesondere solche, deren Filmmaterial Mpeg-2 oder Mpeg-4⁵ kodiert sind auf ihren Bandbreitenbedarf beim Streamen über das Netzwerk untersucht, da diese weit verbreitet sind. Andere Formate, wie z.B. real media liegen in ihrem Bandbreitenbedarf bei gleicher Qualität meist zwischen Mpeg-2 und Mpeg-4 [8].

4.1 Bandbreitenbedarf

Mpeg-4 kodierte Filme benötigen bei DVD-Qualität bis zu 2 MBit Bandbreite. Allerdings erfordert Mpeg-4 zum Abspielen eine relativ hohe Rechenleistung beim Client. Moderne PCs sind jedoch in der Lage Filme auch bei hohen Auflösungen ruckelfrei abzuspielen. Mpeg-4 ist in der Kompressionsrate sehr flexibel. Es ist durchaus möglich Filme bei Datenraten unter 1 MBit zu kodieren, wenn bei der Bildqualität oder der Auflösung Abstriche gemacht werden. Auf diese Weise können auch deutlich mehr Clients pro 100 MBit Link bedient werden. Annehmbare Bildqualitäten lassen sich auch bei Datenraten um die 750 KBit/s erreichen [8]. Diese erreichen jedoch nicht die Qualität von DVDs. Macht man Abstriche bei der Auflösung lassen sich die entstehenden Filme auch von Rechnern mit deutlich geringerer Rechenleistung dekodieren.

Da DVDs und DVB Mpeg-2 kodiertes Filmmaterial bereit stellen, brauchen Mpeg-2 Dateien und DVDs sowie DVB bei gleicher Qualität die selbe Bandbreite. Diese liegt bei handelsüblichen DVDs bei bis zu 8 MBit/s. Über ein 100 MBit Netzwerk lassen sich also etwa 10 Clients gleichzeitig bedienen. DVB-Sender haben Datenraten von 3 MBit/s (DVB-T in Deutschland mit vier Sendern auf einem Kanal) bis theoretisch über 60 MBit/s (DVB-S) [9] [10]. Die Anforderungen an das Netz sind also sehr unterschiedlich. Die benötigte Rechenleistung bei den Clients ist allerdings wesentlich geringer als beim Abspielen von Mpeg-4 Material. Mpeg-2 Dateien lassen sich schon mit Prozessoren bei 400 MHz flüssig abspielen⁶.

³Dieses enthält die Mpeg-2 kodierten Daten in .vob Dateien

⁴Das Kopieren der Inhalte von DVDs auf die Festplatte könnte Urheberrechte verletzen. Des weiteren ist der rechtliche Status der DeCSS Bibliothek, die zum Entschlüsseln verschlüsselter .vob Dateien verwendet wird, umstritten. Das Abspielen verschlüsselter DVDs auf Linux Systemen bewegt sich folglich in einer rechtlichen Grauzone.

⁵Codecs wie XVID oder oDivX

⁶Mindestsystemanforderung auf den meisten mpeg-2 Karten (z.B. für DVB-T)

Grafikkarten mit integrierten Mpeg-2 Dekodern können diesen Wert noch weiter drücken. Auch bei Mpeg-2 kann man die benötigte Bandbreite durch Abstriche in der Qualität senken.

4.2 Streamen von Dateien

In diesem Abschnitt wird beschrieben, wie die einzelnen Front- und Backends zusammen funktionieren.

Apache - VLC

Der VideoLAN Client spielt Dateien, die über HTTP bereit gestellt werden, sofort ab. Das heißt, der VLC verhält sich beim Abspielen von Dateien, die über HTTP bereit gestellt werden, genau so wie bei Dateien, die im lokalen Dateisystem liegen. Wenn der Server HTTP/1.1 unterstützt, kann VLC sogar in den Dateien suchen („spulen“). Der VLC kann keine Playlists verarbeiten, die von einem Webserver bereit gestellt werden. So ist es nur möglich eine Datei pro Aufruf abzuspielen. Das Mozilla Plugin weist dieselben Einschränkungen auf.

Möchte man nur Medien abspielen, die in genau einer Datei kodiert sind, fällt diese Einschränkung nicht ins Gewicht. Möchte man jedoch Filme abspielen, die über mehrere Dateien verteilt sind (z.B. DVD Kapitel, die über mehrere .vob Dateien verteilt sind), ist dies eine wesentlich Einschränkung.

Apache - MPlayer

Der MPlayer spielt Dateien, die über HTTP zur Verfügung gestellt werden, nicht sofort ab, sondern lädt diese zuerst komplett herunter. Dieses ungewollte Verhalten lässt sich allerdings mit einem kleinen Trick umgehen. Wird dem MPlayer eine Playlist im m3u-Format ⁷ zur Verfügung gestellt, spielt dieser sie sofort ab, auch dann wenn sie Dateien referenziert, die auf einem Webserver liegen. Der Player muss jedoch mit dem Argument --playlist aufgerufen werden. Leider kann der MPlayer nicht in Dateien suchen, die über HTTP zur Verfügung gestellt werden.

Mit Hilfe von Playlists könnte der MPlayer auch ganze DVDs abspielen, die als mehrere .vob Dateien auf einem Webserver liegen. Allerdings spielt er immer die erste Tonspur ab, die er findet und bietet keine Unterstützung für DVD Menüs.

Der MPlayer kann zwischen den einzelnen Dateien der Playlist hin- und herspringen. So kann man durch Aufteilen der Filme auf mehrere Dateien Sprungstellen in den Filmen (ähnlich Kapiteln auf DVDs) definieren.

VLS - VLC

Diese vom VideoLAN Projekt vorgeschlagene und favorisierte Lösung hat sich im Test leider als unbrauchbar herausgestellt. Über das loopback Interface verlief das Streamen von DVDs und anderen Medien problemlos. Beim Streamen über ein 100 MBit Netz konnte der Client das Video jedoch nicht flüssig abspielen.

⁷Eine Datei pro Zeile

VLC - VLC

Der VideoLAN Client kann sein Videooutput auch Unicast oder Multicast über das Netzwerk streamen. Ein weiterer VLC kann diesen Stream dann empfangen und abspielen als würde er von einem VLS gesendet. Diese Variante hat sich bei Tests als schnell und stabil auch bei Streaming über das LAN herausgestellt. Allerdings gibt es Probleme mit DVD-Menüs, da diese am Client nicht bedient werden können. Man muss also darauf achten, dass man sofort das richtige Kapitel abspielt. Außerdem muss für jeden Stream eine Instanz des VLC auf dem Server gestartet werden.

4.3 Streamen von DVDs

Wie in den vorherigen Kapiteln beschrieben, kann man DVDs nicht mit voller Funktionalität über HTTP oder andere Protokolle der geprüften Server streamen. Bis hier wurde davon ausgegangen, dass die Clients entsprechend den Anforderungen des Systems konfiguriert sind, die Server jedoch kein Wissen über lokale Gegebenheiten auf den Clients haben müssen. (s. Abb. 1)

Nimmt man von dem letzten Punkt Abstand, können auch ganze DVDs mit voller Funktionalität on Demand bereit gestellt werden. Wie im Abschnitt 2.1 „Die Player“ beschrieben, können beide untersuchten

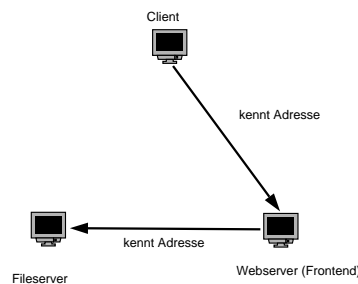


Abbildung 1: Setup bei nicht zentral administrierten Systemen

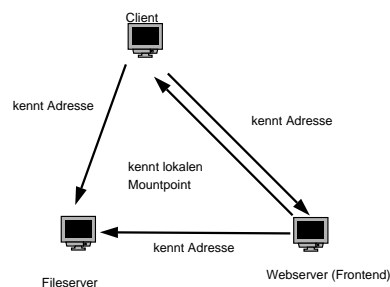


Abbildung 2: Setup für volle DVD Unterstützung

Frontends DVDs, die in ein lokales Verzeichnis kopiert wurden, mit voller Funktionalität abspielen. Lokalen Zugriff auf zentral gespeicherte DVDs kann durch mounten eines Verzeichnisses des Servers auf dem Client erreicht werden. Dies ist per NFS kein Problem. Wird die Oberfläche des Media on Demand Systems zentral bereit gestellt, muss der Administrator den lokalen Mountpoint des DVD-Verzeichnisses kennen, da er Links auf das lokale Dateisystem der Clients in die Oberfläche integrieren muss. Das heißt

auch, dass auf jedem Client das DVD Verzeichnis an der selben Stelle in das Dateisystem eingebunden werden muss. Eine Beispielkonfiguration, die das Streamen von DVDs zulässt, findet sich im Kapitel 5. Diese Lösung ist für Umgebungen, in denen der Administrator keinen Zugriff auf die Clients hat, nicht praktikabel, für zentral administrierte Umgebungen wie z.B. einem Media Zentrum jedoch durchaus umsetzbar. (s. Abb. 2)

4.4 Streamen von DVB

Im Rahmen dieser Arbeit konnte das Streamen von DVB nicht praktisch erprobt werden. Eine Internetrecherche lässt jedoch vermuten, dass das VDR(Video Disc Recorder) Projekt [11] zusammen mit dem Client/Server Plugin [12] die beste Lösung ist, um DVB über ein Netzwerk zu streamen. Das Client/Server Plugin lässt sich über ein Telnet Interface steuern. Als weitere Lektüre wird hier auf die Homepages der einzelnen Projekte hingewiesen. Außerdem gibt es einen umfangreichen Artikel über die Konfiguration von VDR auf einem Einzelplatzrechner in [13].

4.5 Fazit

Das Streamen von einzelnen Dateien wurde von der Kombination Apache-VLC am besten gemeistert, da der VLC in HTTP-Streams suchen kann. Dadurch wird spulen in Filmen möglich. Sollen allerdings mehrere Dateien nacheinander gestreamt werden, wird der MPlayer als Frontend interessant, da dieser Playlists verarbeiten kann.

Die Möglichkeit DVDs von einem VLC zu einem anderen zu streamen ist für eine Media on Demand Lösung schlecht geeignet, da es Probleme mit DVD-Menüs gibt. Das Streaming von DVDs über HTTP ist aus den gleichen Gründen nicht empfehlenswert. Mit einem Workaround (s.o.) ist jedoch auch die volle Unterstützung von DVD-Streaming on Demand möglich. Für diese Anwendung ist der VLC als Player besser geeignet, da er DVD Menüs unterstützt.

5 Beispielkonfiguration einer Media on Demand Lösung

Im Folgenden wird exemplarisch die Konfiguration einer Media on Demand Lösung vorgestellt. Das System unterstützt das Streamen von DVDs, Filmen in diversen Formaten, sowie Musikdateien (hier auch optional Alben, bereitgestellt als Playlist im m3u-Format).

Kiosk Umgebungen lassen sich besonders einfach auf HTML-Basis implementieren. Daher ist auch die hier beschriebene Bedieneroberfläche HTML-basiert. Die hier beschriebene Anwendung ist jedoch kein fertiges Kiosk System. Genaue Anleitungen das Einrichten einer browsergestützten Kiosk-Umgebung unter Linux betreffend finden sich im Internet z.B. unter [5] oder [6]. Das Einrichten einer Kiosk Umgebung ist jedoch nicht Inhalt dieser Studienarbeit. Im Folgenden wird lediglich beschrieben, wie die Server und Clients konfiguriert werden müssen, um einen reibungslosen Betrieb zu ermöglichen.

5.1 Grundsätzliches Setup

Da keines der beiden Frontends alle oben genannten Anforderungen erfüllt, kommen in diesem Beispiel beide Player zum Einsatz. Filme, die in einer Datei vorliegen, sowie mp3-Dateien und Album-Playlists werden mit dem MPlayer wiedergegeben, DVDs werden mit dem VLC abgespielt, da dieser Unterstützung für DVD Menüs bietet. Da dem VLC ein Verzeichnis zum Abspielen übergeben werden muss und dies in HTML schwierig zu realisieren ist, wenn man den VLC als externe Anwendung ansprechen möchte, kommt hier das VLC-Mozilla-Plugin zum Einsatz.

Alle Dateien (außer den DVDs), sowie die Bedieneroberfläche werden von einem Webserver bereit gestellt. Die DVDs werden über NFS⁸ in das Dateisystem der Clients an der Stelle \$MOUNTPOINT eingebunden.

Das Netz, in dem die Clients und Server stehen, ist IP basiert. In diesem Beispiel werden Adressen im Subnetz 10.0.0.0/24 vergeben. Der NFS-Server hat die IP Adresse 10.0.0.1, der Webserver die Adresse 10.0.0.2. Eine Namensauflösung via DNS kann natürlich auch eingerichtet werden, darauf wird allerdings an dieser Stelle nicht eingegangen. Interessierte Leser finden eine ausführliche Anleitung im DNS-HOWTO des Linux Documentation Projects [14].

5.2 Konfiguration

NFS-Server

Der NFS Server soll so konfiguriert werden, dass er das Verzeichnis, in dem sich die Kopien der DVDs befinden für das gesamte Subnetz 10.0.0.0/24 lesend zur Verfügung stellt. Schreibzugriff wird nicht gewährt, damit die Benutzer an den Clients nicht Teile oder ganze DVDs löschen. Die video_ts Verzeichnisse der DVDs liegen auf dem Server im lokalen Verzeichnis /video/DVD/DVDName/.

Der Kernel NFS-Server wird über die Datei /etc/exports konfiguriert. Da der Kernel NFS-Server sehr mächtig ist, wird hier die genaue Syntax der Datei nicht beschrieben. Eine umfangreiche Dokumentation

⁸Network File System, Protokoll für Dateifreigabe unter Linux

des NFS-Servers findet sich in [4] Im folgenden Beispiel einer exports Datei wird ein Verzeichnis (/video/DVD) für ein Subnetz (10.0.0.*) nur lesend (ro) zur Verfügung gestellt. Wichtig ist, dass zwischen dem * und der öffnenden Klammer für die Zugriffsrechtsdefinition kein Leerzeichen stehen darf.

```
# /etc/exports für ein Video on Demand System
#
# Freigabe des DVD-Verzeichnisses für lokale Clients
/video/DVD 10.0.0.*(ro)
#
# end of /etc/exports
```

Webserver

Auf die genaue Konfiguration des Apache-Webservers wird hier aus den selben Gründen wie bei dem NFS-Server nicht näher eingegangen. Im Weiteren wird jedoch davon ausgegangen, dass das Wurzelverzeichnis des Webservers /srv/www/htdocs ist. (Voreinstellung bei SuSE 9.0). Außerdem wird im Folgenden davon ausgegangen, dass die Videos, die nicht als DVD vorliegen, unter /srv/www/htdocs/video liegen. Außerdem wird zu jedem Video eine Playlist im m3u-Format angelegt. Playlists im m3u-Format sind Textdateien, die eine Datei pro Zeile referenzieren. In diesem Beispiel müssen leider absolute Pfade auf dem Webserver referenziert werden. Die Audio-Dateien, sowie Playlists für Alben, werden unter /srv/www/htdocs/audio abgelegt. Eine Beispielplaylist für ein Album mit den Songs Track1.mp3, Track2.mp3 und Track3.mp3 sieht folgendermaßen aus:

```
http://10.0.0.2/audio/Track1.mp3
http://10.0.0.2/audio/Track2.mp3
http://10.0.0.2/audio/Track3.mp3
```

Solche Playlists können auch serverseitig mit Techniken wie PHP oder Perl automatisch generiert werden.

Client

Auf dem Client muss der Mozilla Browser, VLC, das Mozilla-VLC-Plugin sowie der MPlayer installiert werden (s.h. Abschnitt Installation). Außerdem muss das per NFS freigegebene DVD-Verzeichnis in das lokale Dateisystem an der Stelle \$MOUNTPOINT eingebunden werden.

Eine NFS-Freigabe kann man von der Kommandozeile mit dem mount Befehl einbinden. Für dieses Beispiel kann man die folgende Befehlszeile verwenden:

```
root@linux > mount -t nfs 10.0.0.1:/video/DVD $MOUNTPOINT
```

Natürlich muss man \$MOUNTPOINT durch ein beliebiges (existierendes, leeres) Verzeichnis auf dem Client ersetzen. Alternativ kann man das Verzeichnis auch beim Booten des Clients automatisch in das lokale Dateisystem einbinden lassen. Hierzu muss man die Datei /etc/fstab editieren. Ihr muss folgende

Zeile hinzugefügt werden:

```
10.0.0.1:/video/DVD $MOUNTPOINT NFS defaults 0 0
```

Danach muss der Mozilla Browser derart konfiguriert werden, dass er Playlists mit dem MPlayer öffnet. Dazu ordnet man dem MIME-Type audio/mpegurl bzw. audio/x-mpegurl (Dateiendung .m3u) den Befehl `mplayer -fs --playlist` zu⁹. Dies öffnet den MPlayer im Vollbild Modus ohne Bedienelemente, wenn jemand auf einen Link klickt, der zu einer Playlist führt. Sollen die Bedienelemente ebenfalls angezeigt werden, muss der Aufruf zu `gmplayer -fs --playlist` geändert werden.¹⁰ Jetzt ist das Client System fertig konfiguriert.

5.3 Betrieb

Als Bedieneroberfläche und zur Verwaltung der angebotenen Filme oder Alben können beliebige Webseiten im Verzeichnis `/srv/www/htdocs` des Webservers zur Verfügung gestellt werden. Um ein Video bzw. ein Audio-Album abzuspielen, muss einfach ein Link auf die entsprechende Playlist im m3u-Format gelegt werden. Um eine DVD abzuspielen, verwendet man das Mozilla-VLC-Plugin. Wie das Plugin in eine Seite eingebettet wird, zeigt das folgende Beispiel. Es basiert auf einem der Beispiele, die in der Dokumentation des VLC-Players unter [1] zu finden sind.

⁹MIME-Typen dienen dazu verschiedene Klassen von Dateien zu unterscheiden. Typische Beispiele wären Archive (.zip, .gz, .cab, ...) oder mpeg Dateien (.mpg, .mp2, .mp3, .mpeg). Diesen kann man in vielen Browsern bestimmte Programme zum öffnen zuweisen. Z.B. öffne Dateien vom (MIME)-Typ mpeg mit dem MPlayer

¹⁰Möchte man auch Dateien, die vom MPlayer wiedergegeben werden, in eine Webseite einbetten, kann man hierzu den Mozplugger benutzen. Die Installation wird im gleichnamigen Abschnitt dieser Arbeit beschrieben. In diesem Fall können die Bedienelemente allerdings nicht eingeblendet werden.


```

<html>
<head><title>Demo VLC Plugin</title></head>

<body>

<h1>Demo VLC Plugin Beispiel eins: HTTP</h1>

<embed type="application/x-vlc-plugin"
      name="video1"
      autoplay="yes" loop="no" width="400" height="300"
      target="dvd:$MOUNTPOINT/Shrek/video_ts" />
<br />
  <a href="javascript:;"
    onclick='document.video1.play() '>Play video1</a>
  <a href="javascript:;"
    onclick='document.video1.pause() '>Pause video1</a>
  <a href="javascript:;"
    onclick='document.video1.stop() '>Stop video1</a>
  <a href="javascript:;"
    onclick='document.video1.fullscreen() '>Fullscreen</a>

</body>
</html>

```

Die in diesem Beispiel vorgestellten Javascript Befehle zur Steuerung des Plugins (play(), pause(), stop(), fullscreen()), sind alle vom Plugin unterstützten Steuerbefehle. Mehr Kontrolle über den Player kann über Tastaturshortcuts erreicht werden.

Da die Bedienoberfläche komplett HTML basiert ist, ist ihr Erscheinungsbild natürlich sehr flexibel. Eine Bedienoberfläche, die von einem Standardbrowser dargestellt wird, sollte von einem Internet Erfahrenen Benutzer intuitiv bedienbar sein. Selbst Anweisungen zur Konfiguration des Clients können auf einleitenden Seiten der Oberfläche gegeben werden. Da der MPlayer und der VLC auch für Windows Systeme zur Verfügung stehen, ist solch System wahrscheinlich auch plattformübergreifend Einsetzbar.

6 Fazit und Ausblick

Wie in dieser Arbeit gezeigt, ist es mit aktuellen Open Source Programmen möglich, ein Media on Demand System zu implementieren. Wird solch ein System zentral administriert, wie z.B. in Medienzentren, können auch DVDs mit voller Funktionalität zur Verfügung gestellt werden. Die Konfiguration der hier vorgestellten Programme sollten einer Person mit etwas Linux Kenntnis wenig Probleme bereiten. Die Installation verlief auf einem Apt-basierten System einfacher als auf einem RPM-basierten (s.h. Appendix). Einem erfahrenen Linux Administrator sollte die Installation auf beiden Systemen jedoch keine großen Probleme bereiten.

Die Anforderungen an die clientseitig verwendete Hardware sind moderat. Moderne PCs sind leicht in der Lage, die potentiellen Inhalte eines MoD Systems abzuspielen. Sollen die Server nur Inhalte bereit stellen, ist die Anforderung an die Rechenleistung gering - es wird lediglich genügend Festplattenplatz benötigt. Müssen die Server jedoch noch Signale (wie z.B. DVB Sender) umkodieren oder auch zum Mpeg-4 kodieren von Filmen verwendet werden, steigt die Rechenlast rapide. Es sollten dann schnelle Prozessoren eingesetzt werden. Die Anforderungen an die Netzwerkinfrastruktur richtet sich stark nach den Inhalten, die bereit gestellt werden sollen. Sollen nur Inhalte mit geringen Bitraten (wie z.B. Mpeg-4 kodierte Filme oder Musik) gestreamt werden, reicht ein 100 MBit Netzwerk durchaus aus, um kleine bis mittlere Netzwerke zu bedienen. Sollen jedoch zunehmend DVDs oder DVB-Sender über das Netzwerk bereit gestellt werden, müssen die Netzwerkknoten (Switches) und Anbindungen der Server schnell genug ausgelegt werden, um den zusätzlichen Verkehr verarbeiten zu können.

Die in dieser Arbeit vorgestellten Programme sind dauerhaft in der Entwicklung, die jeweiligen open source Projekte sehr aktiv. Es ist zu erwarten, dass Defizite, die in dieser Arbeit aufgedeckt wurden in zukünftigen Versionen bereits behoben sind. Das Streamen von statischen Inhalten von einer Festplatte (Filme ohne Menüführung oder Musik) kann schon jetzt effektiv und stabil implementiert werden. Im Bereich des Streamings von DVDs und insbesondere DVB ist in den kommenden Monaten mit viel Bewegung zu rechnen. Hier lohnt es sich, jeweils die aktuellsten Versionen der Programme zu verwenden.

Literaturverzeichnis

- [1] *Homepage des VideoLAN Projektes*. <http://www.videolan.org>.
- [2] *Homepage des MPlayer Projektes*. <http://www.mplayerhq.hu>.
- [3] Thomas Rohde Lars Eilebrecht, Nikolaus Rath. *Apache Webserver: Installation, Konfiguration, Administration*. mitp, 4th edition, 2002.
- [4] Ricardo Labiaga Hal Stern, Mike Eisler. *Managing NFS and NIS*. O'Reilly & Associates, 2nd edition, 2001.
- [5] *Kiosk-HOWTO*. <http://www.opennet.ru/docs/HOWTO/Kiosk-HOWTO-3.html>.
- [6] *Mozilla Kiosk Projekt*. <http://kiosk.mozdev.org>.
- [7] *Homepage des mozplugger Projektes*. <http://mozplugger.mozdev.org>.
- [8] Dr. Volker Zota. *Kompressionisten - Aktuelle Video-Codex im Vergleich*. *C't*, 10/2003:146, 2003.
- [9] European Telecommunications Standards Institute. *EN 300 744: Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for digital terrestrial television*.
- [10] European Telecommunications Standards Institute. *EN 300 421: Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for 11/12 GHz digital satellite services*.
- [11] *Video Disc Recorder*. <http://www.cadsoft.de/vdr>.
- [12] *VDR Client/Server Plugin Homepage*. <http://www.magoa.net/vdr>.
- [13] Peter Siering. *Fernseh-PC - PC als Videorecorder: c't-VDR-Projekt*. *C't*, 20/2003:174, 2003.
- [14] *DNS-Howto des Linux Documentation Projects*. <http://www.tldp.org/HOWTO/DNS-HOWTO.html>.
- [15] *Debian Homepage*. <http://www.debian.org>.
- [16] *DivX Homepage*. <http://www.divx.com>.
- [17] *XVid Homepage*. <http://www.xvid.org>.
- [18] *Sourceforge.net LAME Projekt*. <http://sourceforge.net/projects/lame/>. Lame Aint an MP3 Encoder.

A Installation eines Debian Systems

Die Installation der Komponenten auf einer Debian Testing/Unstable Distribution gestaltet sich einfach, da alle Programme bereits vollständig kompiliert in apt Repositories existieren. So können sie mit dem apt Paketmanager installiert werden.

Im Folgenden wird die Installation eines kompletten Systems mit Hilfe eines Netinst Images für Debian Testing (Sarge) beschrieben. Dieses kann man unter [15] beziehen. Für die Installation sollte man sich die Betriebsdaten seines Monitors (Horizontale und Vertikale Bildwiederholraten, sowie Auflösung) zurecht legen. Dies ist die einzige Einstellung, die manuell vorgenommen werden sollte. Alle anderen Komponenten wurden auf dem verwendeten Testsystem automatisch erkannt.

Die Grundkonfiguration und Installation des Betriebssystems gestaltet sich einfach. Man sollte bei den Installationsaufgaben eine grafische Benutzeroberfläche auswählen. Außerdem ist es angebracht die Unix Server Komponenten (insbesondere den Kernel NFS Dämon) zu installieren.

Nach der Installation muss man noch einige apt-Repositories zu seiner Systemkonfiguration hinzufügen. Dazu editiert man die Datei /etc/apt/sources.list und fügt ihr folgende Einträge hinzu:

```
# Zusätzliche Debian Repositories

# Debian Unstable
deb http://debian.tu-bs.de/debian unstable contrib
deb-src http://debian.tu-bs.de/debian unstable contrib

# MPlayer Repository
deb http://debian.tu-bs.de/mplayer unstable

# VLC Repository
deb http://www.videolan.org/pub/videolan/debian sid main
```

Danach kann führt man ein Update auf Debian Testing/Unstable durch:

```
root@linux > apt-get update
root@linux > apt-get dist-upgrade
```

Danach hat man ein lauffähiges GNU/Debian unstable/testing installiert.

Die verwendete Software lässt sich mit dem apt Paketmanager installieren. Um ein Programm zu installieren führt man den folgenden Befehl aus:

```
root@linux > apt-get install Paketname
```

Abhängigkeiten werden vom apt-System aufgelöst und eventuell zusätzlich benötigte Komponenten, wie z.B. Bibliotheken werden automatisch installiert.

Die Apt-Pakete zu den in dieser Arbeit besprochenen Programmen heißen wie folgt.

- MPlayer (optimierte Pakete für i386/i686/k7 (Athlon) Systeme): mplayer-{386|686|k7}
- MEncoder (zum kodieren von Filmen): mencoder
- VideoLAN Client (VLC): gnome-vlc
- VideoLAN Server (VLS): vls
- Unterstützung für verschlüsselte DVDs im VLC: libdvdcss2
- VLC Mozilla-Plugin: mozilla-plugin-vlc
- Apache Webserver: apache
- Mozplugger: mozplugger¹¹

Das folgende Shell-Script installiert alle Pakete.

¹¹hier wird auch eine umfangreiche Beispielkonfigurationsdatei unter /etc installiert

```
#!/bin/bash
# Shell Script für die Installation der Softwarekomponenten
# eines Media on Demand Systems
# Autor Steffen Stein
# Dieses Script muss als root ausgeführt werden!

# Systemarchitektur Eintragen für spezialisierten MPlayer
# mögliche Werte: i386 i686 k7
# Default i386, da Kompatibel zu allen anderen
ARCH=i386
# Installation MPlayer
apt-get -y install mplayer-$ARCH
# Installation des MEncoders
apt-get -y install mencoder

# Installation der VideoLAN Komponenten
# Client mit Unterstützung von verschlüsselten DVDs
apt-get -y install gnome-vlc libdvdcss2
# Server
apt-get -y install vls
# Mozilla Plugin
apt-get -y install mozilla-plugin-vlc

# Apache Webserver
apt-get -y install apache

# Mozplugger
apt-get -y install mozplugger

# End of Script
```

B Installation eines SuSE Linux Systems

Im Folgenden Abschnitt wird die Installation von SuSE Linux 9.0 zuzüglich aller Komponenten, die in dieser Studienarbeit erwähnt wurden, beschrieben. Hierbei wird nicht auf die grundlegende Konfiguration von SuSE eingegangen, sondern nur eine Paketauswahl, die funktioniert, beschrieben.

Ausgehend von einem „minimalem grafischen System“ müssen noch einige Pakete per YaST installiert

werden, damit alles weitere funktioniert:

- automake
- autoconf
- make
- gcc
- gcc-c++
- libdvread und libdvread-devel
- libdvplay und libdvplay-devel
- libdv
- libvorbis und libvorbis-devel
- vorbis-tools
- mad und mad-devel
- alsa-devel
- asound und asound-devel
- arts und arts-devel
- SDL und SDL-devel
- unzip
- Realplayer
- mozilla und mozilla-devel
- mozplugger
- XFree86-devel
- gtk2-devel
- freetype
- kdelibs3
- XFree86-compat-libs
- lirc
- gnome-libs
- desktop-file-utils
- xview
- flac und flac-devel

Natürlich kann man noch diverse andere Pakete, wie z.B. KDE als Desktop installieren. Die jeweiligen Abhängigkeiten sollte man von YaST auflösen lassen.

Nachdem die Installation durchgelaufen ist, muss man noch das Mozplugger Plugin an die richtige Stelle kopieren:

```
root@linux > cp /usr/lib/browser-plugins/mozplugger.so \  
/opt/mozilla/lib/plugins/
```

Um Kompatibilität mit den am häufigsten verwendeten Codecs zu erreichen sollte man noch DivX und XVID installieren. Erstere können unter [16] heruntergeladen werden. Nachdem man DivX für Linux heruntergeladen und das Paket dekomprimiert hat, installiert man den Codec per

```
root@linux > ./install.sh
```

im entstandenen Verzeichnis.

Die aktuellen Quellen des Xvid-Codecs bekommt man unter [17]. Nach dem Herunterladen und Auspacken installiert man die Codecs mit

```
root@linux > cd build/generic
root@linux > ./configure
root@linux > make
root@linux > make install
```

Die aktuellen Quellen des lame-Projekts (zum Komprimieren im mp3 Format) findet man im Internet unter [18]. Das Paket muss man auspacken und kann es dann per

```
root@linux > ./configure --prefix=/usr
root@linux > make
root@linux > make install
```

installieren.

MPlayer

Den Quelltext des MPlayer kann man im Internet unter [2] herunterladen. Damit man auch die grafische Oberfläche nutzen kann, sollte man sich auch einen Skin, sowie eine Font (für das On screen Display) herunterladen.

Nachdem man den Quelltext des MPlayer ausgepackt hat, wird er folgendermaßen kompiliert und installiert:

```
user@linux > ./configure --enable-gui --language=de
user@linux > make
root@linux > make install
```

Man beachte, dass sowohl configure als auch make als normaler user ausgeführt werden müssen, da sonst einige Bibliotheken nicht richtig eingebunden werden. Den Skin entpackt man nach `/usr/share/mplayer/Skin/default`, die Font nach `/usr/share/mplayer/font`.

Danach sollte sich der MPlayer problemlos starten lassen. Eine ausführliche Beschreibung aller Fähigkeiten des MPlayers findet sich in der manpage:

```
user@linux > man mplayer
```

Außerdem wurde nebenbei noch der mencoder installiert. Dies ist ein sehr leistungsfähiges Tool zum umcodieren aller Formate, die der MPlayer abspielen kann. Auch hier findet sich ausführliche Dokumentation in der manpage.

Des Weiteren findet sich die komplette Dokumentation des MPlayers in neun verschiedenen Sprachen unter [2].

VLC

Auf der Homepage des VideoLAN-Projektes [1] finden sich RedHat 9 RPMs, sowohl für VLC-Player als auch für die benötigten Bibliotheken. Da RedHat und SuSE Linux beide eine rpm basierte Paketverwaltung verwenden, sind diese Dateien auch für die Installation auf einem SuSE System geeignet.

Damit man den Videolan-Client installieren kann, muss man sich noch folgende Pakete von der Homepage herunterladen und installieren:

- imlib2
- ebml
- lame (um eine Fehlermeldung zu vermeiden)
- faad2
- matroska
- openslp
- wxGTK
- libdvbpsi
- libtar
- ffmpeg
- xvidcore
- xosd
- libdvdcss
- a52dec
- mpeg2dec

Die Pakete werden per

```
root@linux > rpm -Uhv Paketname.rpm
```

installiert. Hierbei muss man lediglich beachten, dass man das Paket imlib vor dem Paket ffmpeg installiert.

Es gibt eine Inkompatibilität zwischen RedHat und SuSE. Das Paket, das die mad-library zur Verfügung stellt heißt bei ersteren libmad, beim zweiten mad. Das resultiert in einem Fehler wegen nicht erfüllter Abhängigkeiten, wenn man versucht das VLC-RPM zu installieren. Man kann das Paket trotzdem per

```
root@linux > rpm -Uhv --nodeps VLC-xxx.rpm
```

installieren. Wenn man das software Plugin von YaST das nächste mal startet, wird es eine Fehlermeldung wegen nicht erfüllter Abhängigkeiten, die YaST nicht auflösen kann, geben. Hier kann man bei allen Punkten „ignorieren und ein inkonsistentes System riskieren“ anwählen.

Apache

Den Webserver kann man per YaST installieren. Die entsprechende Auswahl heißt „einfacher Webserver mit Apache.“ Je nachdem, ob der Server auch eine Webseite hosten soll, die Techniken wie php oder perl benutzt, müssen diese Module auch installiert werden.

VLS

Um VLS zu installieren muss man sich das Quelltextpaket von [1] herunterladen. Das Kompilieren und Installieren verläuft folgendermaßen:

```
root@linux > /configure --enable-dvb \  
--with-libdvb=/usr/local/lib --with-dvb=/usr/local/include  
root@linux > make  
root@linux > make install
```

Um DVB Unterstützung beim VLS zu erreichen muss man vorher per YaST die DVB Header Dateien installieren.