

Linux & Audio

Linux Audiosysteme und deren Verwendung

Christoph Eckert

Graf-Rhena-Straße 2

76137 Karlsruhe, Germany

mchristoph.eckert@t-online.de

März 2005

Übersicht

In den letzten Jahren wurden die Audiofähigkeiten von linuxbasierten Betriebssystemen immens verbessert. Für fast alle Audiobelange, insbesondere denen professioneller Musiker, sind inzwischen freie Softwareprodukte verfügbar.

Andererseits führten die rapiden Umbrüche auch zu einem derzeit noch komplexen und nicht ganz einfach zu handhabenden System. Mit dem richtigen Hintergrundwissen kann man fast alle Aufgaben bewältigen – oft sogar leichter als auf anderen Systemen. Das hierzu wichtigste Basiswissen versucht dieses Papier zu vermitteln.

1 Einleitung

In den letzten Jahren hat sich im Bereich Linux und Audio einiges getan. Insbesondere im Bereich professioneller Audioverarbeitung fand in den letzten drei Jahren durch das Erscheinen einiger wichtiger Kernkomponenten eine rasante Entwicklung statt. Andererseits ist es für den Durchschnittsanwender noch immer nicht ganz trivial, einfache Dinge, wie beispielsweise mehrere Audioprogramme gleichzeitig laufen zu lassen oder aber eine Schallplatte zu digitalisieren, durchzuführen.

2 Terminologie

Im Zusammenhang mit Audio unter Linux fallen immer wieder einige Begriffe, die verstanden sein wollen, bevor man sich zielstrebig zu Werke begeben kann.

2.1 Treibersysteme

Im Linuxkernel finden sich - historisch bedingt - zwei verschiedene Treibersysteme, das ältere »Open Sound System«^[1] (kurz OSS) und die neuere »Advanced Linux Sound Architecture«^[2] (kurz ALSA). In Kernen der 2.4er Reihe fanden sich im Kernel ausschließlich OSS-Treiber¹. Die

ALSA-Treiber musste man folglich bei den 2.4er Kernen noch separat vom Kernel installieren. Ab den Kernen der 2.6er Reihe sind die ALSA-Treiber neben den OSS-Treibern im Kernel vorhanden. Die OSS-Treiber sind bereits als »Deprecated« markiert, werden somit irgendwann entfernt und sollten daher nicht mehr verwendet werden:



Ab den Kernen der 2.6er-Reihe braucht man also das Paket »alsa-driver« nicht mehr separat zu installieren. ALSA besteht jedoch nicht nur aus einem Paket mit Treibern für verschiedene Audiokarten, sondern auch aus einer Schnittstelle für Audioapplikationen. Daher benötigt man noch das Paket »alsa-lib«, welches diese Schnittstelle beinhaltet. Optional empfiehlt sich weiterhin das Paket »alsa-utils«, welches einige nützliche Hilfsprogramme, wie beispielsweise das Kommandozeilenwerkzeug »alsacnf« zur Konfiguration von Soundkarten, beinhaltet.

Sind diese Pakete installiert und die Soundkarte (beispielsweise unter Zuhilfenahme von alsacnf) eingerichtet, fügt man ALSA noch zum Startvorgang hinzu, damit es beim Rechnerstart automatisch geladen wird².

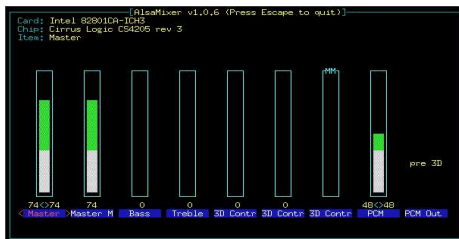
Fast alle aktuellen Distributionen bringen von Haus aus ALSA-Unterstützung mit, wobei die Soundkarten bereits bei der Systeminstallation erkannt und eingerichtet werden. Dies ist beispielsweise bei den Distributionen der Firmen SuSE^[3] und Mandriva^[4] der Fall. Sie brauchen sich in diesen Fällen nicht selbst um die Einrichtung von ALSA zu kümmern. Lediglich bei

Betriebssystemen, die bevorzugt auf Servern zum Einsatz kommen, findet sich oft noch OSS als Treibersystem. Dies trifft beispielsweise auf die »Stable«-Zweige der Debian-Distribution[5] zu. Gentoo[6]-Benutzern steht ein sehr guter Leitfaden zur ALSA-Einrichtung zur Verfügung.

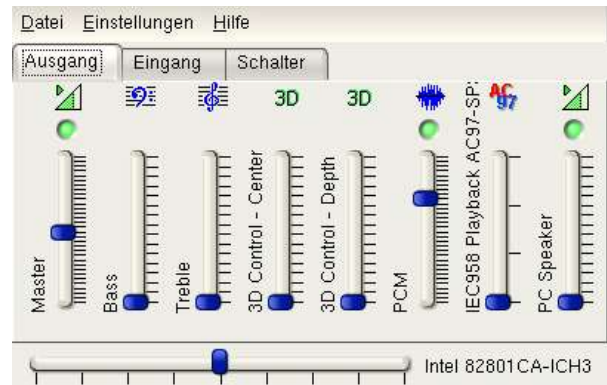
Damit bei der Verwendung von ALSA nicht alle Audioprogramme, die noch auf OSS als Treibersystem setzen, unbrauchbar werden, enthält ALSA eine OSS-Emulation. Das bedeutet, dass ALSA einem OSS-basierten Audioprogramm vorgaukelt, OSS sei vorhanden. Ist diese ALSA-OSS-Emulation im Kernel eingeschaltet, so finden sich auch bei der Verwendung von ALSA nach wie vor die Gerätedateien für OSS, wie beispielsweise /dev/dsp und /dev/midi im System vor. ALSA-basierende Programme verwenden diese Gerätedateien nicht.

2.2 Mixer

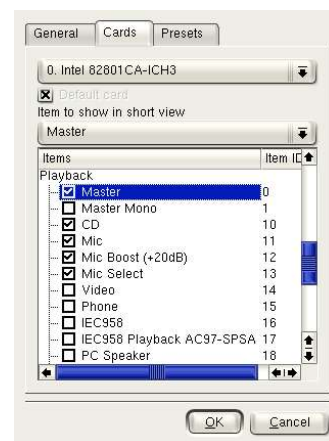
Um das Verhalten einer Soundkarte, wie Aufnahme- und Wiedergabelautstärke etc. steuern zu können, benötigt man ein entsprechendes Programm. Dieses wird als Mixer³ bezeichnet. Mit dem Paket alsa-utils wird beispielsweise der textbasierte »alsamixer« installiert:



Neben alsamixer gibt es auch einige graphische Mixer wie beispielsweise qamix und die beiden KDE-Mixer kmix und kamix, die sich in die Kontrollleiste einbetten. Kmix ist der Standardmixer des KDE-Projektes. kmix kann sowohl bei Verwendung von OSS als auch bei Verwendung von ALSA eingesetzt werden:



Soundkartenmixer sind für den Anwender im Allgemeinen nicht ganz einfach zu verstehen. Dies liegt an mehreren Faktoren. Der ALSA-Treiber dient dazu, den auf der Audiokarte vorhandenen Chip nutzen zu können. Diese Chips sind teilweise wahre Alleskönner, die neben der eigentlichen Audiokartenfunktionalität auch Modemfunktionen zur Verfügung stellen, sich zum Anschluss eines Telefones eignen oder 3D-Sound erzeugen können. Es kann aber auch sein, dass nicht alle Fähigkeiten des Chips auch tatsächlich an das Computergehäuse herausgeführt sind. Deshalb werden im Mixerprogramm eventuell Bedienelemente für Anschlüsse angezeigt, über die der Rechner gar nicht verfügt. Diese Problematik existiert nicht nur auf linuxbasierten Systemen. Daher bieten manche Mixer die Möglichkeit, die nicht benötigten Bedienelemente auszublenden, wie hier in den Einstellungen von kamix:



Da moderne Karten derart komplex sind, müsste man sich als Anwender eigentlich intensiv mit den Möglichkeiten und dem Audiofluss in der Karte beschäftigen, um den Mixer wirklich verstehen zu können.

2.3 Soundserver

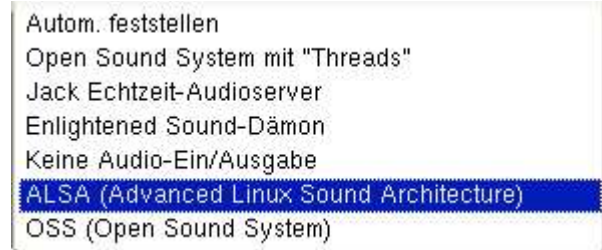
Grundsätzlich ist es jetzt, mit den ALSA-Treibern, der `alsa-lib` und dem Mixer aus dem `ALSA-Utills-Paket`, bereits möglich, dass ein einzelnes Audioprogramm über die Audiokarte Klänge ausgibt und Sie diese wie gewünscht über das Mixerprogramm in der Lautstärke regeln können. Das Audioprogramm, beispielsweise ein Musikwiedergabeprogramm, verbindet sich über die ALSA-Schnittstelle mit der Audiokarte und gibt Klänge aus.

Die Betonung liegt hierbei jedoch auf »einzelnes Programm«. Sobald nämlich ein Programm auf diese Weise die Audiokarte belegt, kann kein weiteres Programm gleichzeitig Audio auf die selbe Schnittstelle ausgeben. Stattdessen wird das zweite Programm entweder eine Fehlermeldung ausgeben, im Hintergrund warten und erst auf dem Bildschirm erscheinen sobald das Audiogerät frei wird oder gar nicht erst starten. Aus diesem Grunde wurde es notwendig, zwischen die Audiokarte und die Audioprogramme eine Zwischenschicht, den sogenannten Soundserver, einzuschieben. Dieser sammelt die Audioausgabewünsche der einzelnen Programme, mischt diese und gibt diese anschließend auf die Audiokarte aus.

Grundsätzlich ist diese Vorgehensweise sehr intelligent. Allerdings existiert nicht ein einzelner allgemeingültiger Soundserver, sondern es gibt mehrere, und auf einer einzelnen Audiokarte kann normalerweise⁴ nur ein einzelner Soundserver laufen, nicht mehrere. Es sei denn, `Dmix` (siehe dort) ist konfiguriert, dann können auch mehrere verschiedene Soundserver gleichzeitig laufen.

2.3.1 aRts

»A Realtime Synthesizer« (kurz `aRts`) wurde ursprünglich tatsächlich als Synthesizer konzipiert. Es stellte sich dabei heraus, dass er über Eigenschaften verfügt, die sich auch als Soundserver nutzen lassen. `aRts` wurde daher vom KDE-Desktop-Projekt[7] als Standardsoundserver verwendet. Alle KDE-konformen Programme geben Audio an `aRts` aus, der dafür sorgt, dass die verschiedenen Klänge zusammengemischt auf die Soundkarte ausgegeben werden können. Dabei kann `aRts` über verschiedene Treiber und weitere Schnittstellen ausgeben, darunter `OSS` und `ALSA`:



`aRts` wird üblicherweise beim Start von KDE mitgestartet und steht von da an allen KDE-Applikationen zur Verfügung. Alle KDE-konformen Programme können somit Klänge ausgeben.

Sobald jedoch ein Programm, welches nicht über `aRts` ausgeben kann, Audio ausgeben möchte, wird dies nicht funktionieren. Das Programm kann nicht über `ALSA` ausgeben, da bereits `aRts` die Audiokarte blockiert, und auch ein zweiter Soundserver wie `esound`, sofern vom zweiten zu startenden Programm benötigt, kann nicht gestartet werden. Zwar können Sie `aRts` über das KDE-Kontrollzentrum auch abschalten, woraufhin die Audiokarte für andere Programme frei wird, aber die meisten KDE-Programme werden ab diesem Zeitpunkt ihrerseits nicht mehr in der Lage sein, Audio auszugeben⁶.

Zu `aRts` bleibt anzumerken, dass dessen Weiterentwicklung eingestellt ist. Daher sucht das KDE-Projekt für eine der nächsten Versionen ihrer Desktopumgebung einen Ersatz. Es gibt dabei erfolgversprechende Bestrebungen, zusammen mit dem GNOME-Desktop-Projekt[8] eine gemeinsame Lösung zu finden, so dass künftig Gnome- und KDE-basierende Audioprogramme gleichzeitig über ein und denselben Soundserver Audiodaten ausgeben können. Ein möglicher Soundserver hierfür könnte `Gstreamer`[16] sein.

2.3.2 Enlightened Sound Daemon

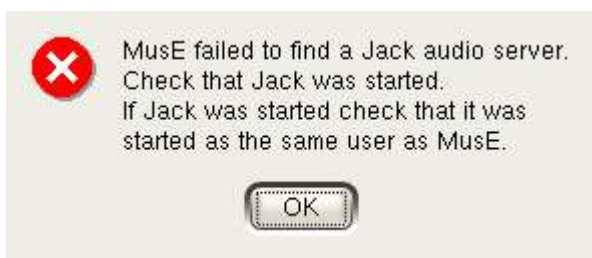
Der »Enlightened Sound Daemon« (kurz `esound`) ist der Soundserver des Gnome-Projektes. Er erfüllt im Wesentlichen die selbe Aufgabe wie `aRts` für das KDE-Projekt. Er mischt die Audioausgabewünsche verschiedener Gnome-Programme und gibt diese über die Audiokarte aus. Auch hier besteht das selbe Problem wie schon bei `aRts`. Programme, die Audio nicht über `esound` ausgeben können, werden entweder nicht starten, eine Fehlermeldung produzieren oder zumindest stumm bleiben. Beendet man `esound`, verliert man den Vorteil des Soundservers und

esound-basierende Programme werden keine Klänge mehr ausgeben können.

Auch für esound gibt es Ersatzbestrebungen, beispielsweise das Polypaudio-Projekt[9] oder der schon bei aRts angesprochene Gstreamer.

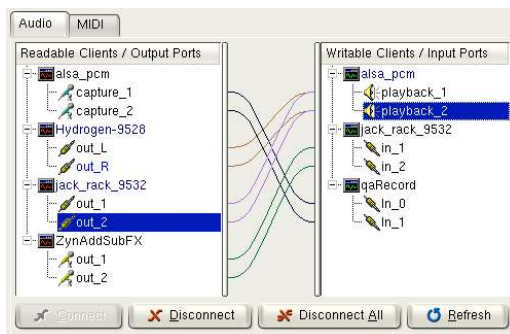
2.3.3 JACK

Das »Jack Audio Connection Kit«[10] beinhaltet als Hauptbestandteil einen speziellen Soundserver (jackd). Im Gegensatz zu aRts oder esound ist JACK auf die Belange professioneller Musiker hin abgestimmt. JACK ist ein echtzeitfähiger Audioserver, der dazu verwendet werden kann, einen linuxbasierten Rechner als virtuelle Studioumgebung zu nutzen. Hierbei wurde besonderer Wert auf geringste Latenzen gelegt, d.h. dass Audioströme möglichst schnell und ohne Unterbrechung durch den Rechner laufen. Dies ist wichtig, da Audiodaten – im Gegensatz zu Videodaten – äußerst timingkritisch sind. Fehlt in einer Videosequenz ein Bild, wird dies meist gar nicht auffallen. Fehlt in einem Audiostrom ein Sample, wird man das sofort hören. Auch ist ein möglichst rascher Transport der Audiodaten notwendig, wenn der Rechner als Effektgerät verwendet werden soll. Schließlich möchte man beim Anreißen einer Gitarrenseite den Klang sofort aus dem Lautsprecher hören und nicht erst drei Sekunden später. Fast alle Audioprogramme aus dem Musikerumfeld verwenden inzwischen JACK als Soundserver und lassen sich ohne laufenden JACK teilweise gar nicht starten, wie beispielsweise die Sequencing-Software MuSE [11]:

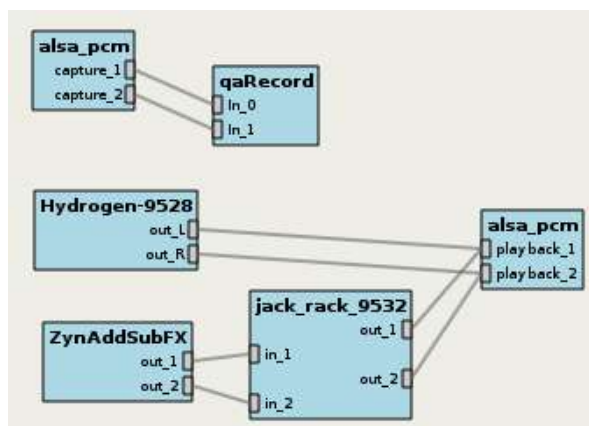


Eine weitere Besonderheit von JACK ist, dass sich JACK-Programme nicht automatisch mit dem Soundserver verbinden. Denn es ist Bestandteil des JACK-Konzeptes, dass Audiodaten nicht nur zur Soundkarte, sondern auch an andere Programme geschickt werden können. So kann ein Softwaresynthesizer seine Audioausgabe zuerst an ein Programm zur Klangverfremdung weiterreichen, und erst danach wird der Klang auf

die Audiokarte ausgegeben. Daher müssen JACK-Programme nach dem Starten vom Anwender manuell miteinander verbunden werden, bevor eine Audioausgabe erfolgen kann. Hierzu verwendet man beispielsweise das Kommandozeilenprogramm »jack-connect« oder eine graphische Alternative wie beispielsweise »qjackctl«[12]...



oder auch »Patchage«[13]:



Einer der Nachteile von JACK besteht darin, dass er dem Benutzer über ein Kernelmodul Echtzeitrechte, das bedeutet eine Bevorzugung des Programmes bei der Zuteilung von Rechenzeit, einräumt. Ein fehlerhaft geschriebenes Audioprogramm, welches auf JACK aufsetzt, kann dadurch unter Umständen den Rechner zum Stillstand bringen, indem es zuviel Rechenzeit anfordert. Zwar kann man JACK auch ohne diese Echtzeitprivilegien laufen lassen, allerdings erhöhen sich dadurch wieder die Latenzen, so dass die Audiodaten beim Weg durch den Rechner wieder ausgebremst werden.

Durch Weglassen der Echtzeitpriorität könnte JACK zu einem allgemeinen Audioserver für Linux werden. Allerdings liegt der

Entwicklungsfocus derzeit eben auf Aspekten aus dem Musiker- und Studioumfeld.

2.3.4 ALSA mit Dmix-Plugin

Für ALSA existieren derzeit drei Plugins[14], die sich zwischen die ALSA-Schnittstelle und den eigentlichen Audiokartentreiber einklinken lassen. Darunter findet sich auch das Dmix-Plugin. Das Dmix-Plugin gestattet es mehreren Audioprogrammen gleichzeitig, Audio auszugeben, indem es die verschiedenen Audioströme zusammenmischt und auf die Soundkarte ausgibt (auch als »Softwaremixing« bekannt). Strenggenommen ist Dmix kein Soundserver im klassischen Sinne, aus Anwendersicht verhält sich Dmix jedoch ganz ähnlich.

Derzeit findet man Dmix auf nur wenigen Systemen vorkonfiguriert. Dies kommt daher, dass Dmix einerseits eine bisher wenig bekannte Entwicklung ist, die Einrichtung von Dmix andererseits derzeit noch durch das manuelle Erstellen einer komplexen Konfigurationsdatei⁷ erfolgen muss. Es sieht jedoch danach aus, dass Dmix in naher Zukunft von den großen Distributionen standardmäßig installiert werden wird. Es bleibt zu hoffen, dass die Entwickler von audioausgebenden Programmen in ihre Applikationen möglichst rasch JACK- und ALSA-Unterstützung einbauen, so dass diese standardmäßig via JACK (für die Musiker) und Dmix (für die Normalanwender) Audio auszugeben in der Lage sind.

2.3.5 MAS, Gstreamer etc.

Die bisher angesprochenen Systeme beschränken sich vornehmlich darauf, Audio lokal auf einem einzelnen Rechnersystem zu verarbeiten. Es gibt jedoch weitere Ansätze, die sich darauf fokussieren, Audioströme nicht nur lokal auf dem Rechner, sondern generell netzwerkweit zur Verfügung zu stellen. Sowohl der »Media Application Server«[15] (kurz MAS) als auch »Gstreamer«[16] gehören zu dieser Gattung. Gstreamer wird derzeit als Kandidat angesehen, die beiden Soundserver esound und aRts zu ersetzen. Sowohl MAS als auch Gstreamer sind derzeit noch nicht weit verbreitet und befinden sich noch in einem frühen Entwicklungsstadium.

2.4 Duplex und Full-Duplex

Diese beiden Begriffe sind etwas verwirrend. Eine Audiokarte ist »duplex«-fähig, wenn diese sowohl Audio aufnehmen als auch abspielen kann, nicht jedoch gleichzeitig. Erst dann, wenn man über eine Audiokarte gleichzeitig aufnehmen und abspielen kann, heißt diese »full-duplex«-fähig. Die meisten der heute erhältlichen Audiokarten sind fullduplex-Karten. Dies gilt selbst für die On-board-Karten nach dem AC '97-Standard. Eine Einschränkung gibt es jedoch auch bei diesem Standard. Man kann zwar gleichzeitig eine Radiosendung aufnehmen und dabei seine Lieblingsstücke über einen Musikabspieler anhören, man kann aber niemals gleichzeitig von verschiedenen Quellen aufnehmen, also beispielsweise während der Digitalisierung der Schallplattensammlung auch noch eben mal schnell eine Radiosendung mitschneiden. Selbiges gilt für das Abspielen. Man kann nicht auf einem Lautsprecherpaar Musik hören (für das Wohnzimmer beispielsweise) und auf einem weiteren Lautsprecherpaar Radio hören (Küche oder Badezimmer).

2.5 Audioabstraktionsebenen

Audioabstraktionsebenen sind spezielle Bibliotheken, die es den Programmierern eines Audioprogrammes gestatten, die Applikation plattformunabhängig zu schreiben. Eine der bekannteren Audioabstraktionsebenen ist »Portaudio«[17], welches beispielsweise von »Audacity«[18] verwendet wird. Portaudio schiebt sich zwischen die Audioapplikation und das darunterliegende Betriebssystem, so dass sich die Programmierer der Audioapplikation nicht um die Besonderheiten des darunterliegenden Systemes kümmern müssen. Audioapplikationen, die Portaudio verwenden, sind daher mit relativ geringem Aufwand sowohl unter Linux, Mac OS oder auch Windows zum Laufen zu bekommen. Portaudio ist jedoch kein Soundserver. Portaudio ist eine Abstraktionsschicht, die es Programmierern erleichtert, ein Audioprogramm plattformunabhängig zu halten. »Portmidi« ist Teil des Portaudioprojektes, um auch MIDI-Applikationen plattformunabhängig gestalten zu können:

Plattformunabhängiges Audio-Programm (z.B. Audacity)
Audio-Abstraktionsebene (z.B. Portaudio)
Linux, Windows oder Mac Audio-System
Betriebssystem (Linux, Windows, Mac OS)
Hardware

Als Anwender sollte man normalerweise von Portaudio überhaupt nichts mitbekommen, es sei denn, man möchte eine auf Portaudio basierende Applikation selbst kompilieren. In diesem Falle muss man die Portaudio Entwicklungsbibliotheken installiert haben.

3 Anwendung

Nachdem die grundsätzlichen Begriffe eines Linux-Audio-Systemes bekannt sind, sollen konkretere Beispiele verdeutlichen, worauf man als Anwender achten sollte.

3.1 Audiowiedergabe

Im Optimalfalle würde jedes Audioprogramm in der Lage sein, auf alle bekannten Audiosysteme auszugeben, beim Aufstarten das System analysieren und sich automatisch mit dem verfügbaren Audiosystem, wie aRts, esound, JACK oder auch ALSA-Direktzugriff, verbinden. Dies trifft jedoch auf die meisten Programme nicht zu.

Derzeit bleibt es daher dem Anwender überlassen, sich mit den verschiedenen Audiosystemen auszukennen und sicherzustellen, dass die einzelnen Programme wie gewünscht zusammenarbeiten. Startet man ein Audioprogramm und erhält keine Audioausgabe, sollte man in der Lage sein zu ermitteln, welches Audiosystem gerade läuft, ob das gestartete Programm damit zusammenarbeiten kann und ob die Audiokarte korrekt eingestellt ist.

Zu diesem Zweck kann es sehr nützlich sein, ein Programm, welches nicht wie erwartet reagiert, aus einem Terminalfenster heraus zu starten. Oft liefern die Ausgaben des Programmes im Terminalfenster wertvolle Hinweise auf die Fehlerursache:

```
bash-2.05b# xmms Titel-1.mp3
Message: fmt 5, channels: 2

** WARNING **: alsa_setup(): Failed to open pcm device (hw:0.0):
Das Gerät oder die Ressource ist belegt
[]
```

Ferner kann man sehr vielen Programmen über das Terminalfenster Informationen über die Audiosysteme, die es unterstützt, entlocken, indem man das Programm mit bestimmten Parametern aufruft. Im einfachsten Falle sollten die Parameter -h bzw. --help weitere Auskünfte erteilen. mplayer bietet sogar einen eigenen Aufruf, um die unterstützten Audiosysteme anzuzeigen:

```
MMX supported but disabled
MMX2 supported but disabled
SSE supported but disabled
bash-2.05b# mplayer -ao help
MPlayer 1.0pre5try2-3,3,4 (C) 2000-2004 MPlayer Team

CPU: Intel Pentium 4/Xeon/Celeron Foster 1197 MHz (Family: 8, Stepping: 4)
Detected cache-line size is 64 bytes
MMX supported but disabled
MMX2 supported but disabled
SSE supported but disabled
CPUflags: MMX: 0 MMX2: 0 3DNow: 0 3DNow2: 0 SSE: 0 SSE2: 1
Compiled for x86 CPU with extensions: SSE2

Reading config file /usr/share/mplayer/mplayer.conf
Reading config file /home/ice/mplayer/config
Available audio output drivers:
  mpegpes Mpeg-PES audio output
  oss      OSS/ioctl audio output
  alsa     ALSA-0.9,x-1,x audio output
  arts     aRts audio output
  esd      Esound audio output
  jack     JACK audio output
  sdl      SDLlib audio output
  null     Null audio output
  pcm      RAW PCM/WAVE file writer audio output
  plugin   Plugin audio output
```

Wie man sieht, kann mplayer mit einer Vielzahl verschiedener Audiosysteme umgehen.

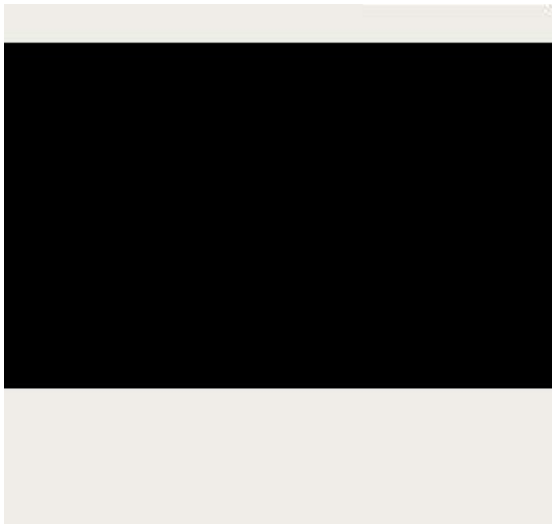
3.1.1 Direktzugriff auf die Audiokarte

Greift ein Programm via Direktzugriff auf das Audiogerät zu, dann ist, wie bereits angesprochen, währenddessen das Audiogerät in aller Regel für andere Programme blockiert⁴.

Dabei kann es vorkommen, dass es für den Anwender nicht ganz offensichtlich ist, dass das Audiogerät bereits belegt ist. Dies ist beispielsweise dann der Fall, wenn gegenwärtig zwar keine konkrete Audioausgabe erfolgt und somit nichts zu hören ist, aber dennoch beispielsweise ein im Hintergrund laufender Soundserver wie esound, aRts oder JACK das Audiogerät belegt hat. In diesem Falle muss der Soundserver beendet werden, bevor ein weiteres Programm direkten Zugriff auf die Audiokarte erhalten kann.

Angenommen Sie verwenden KDE. In diesem Falle wird der Soundserver aRts höchstwahrscheinlich im Hintergrund laufen⁵. Sie möchten nun mit dem Realplayer einen Beitrag einer Nachrichtenseite ansehen. Realplayer wird

zwar auf dem Bildschirm erscheinen, aber nicht auf Klicken reagieren oder wie gewünscht das Video abspielen. Ganz im Gegenteil kann es vorkommen, dass dessen Bedienelemente ganz verschwinden:



Erst nachdem Sie über das KDE Kontrollzentrum oder mittels des Befehles »artshell -q terminate« aRts beendet haben, wird Realplayer mit dem Abspielen des Videos beginnen können.

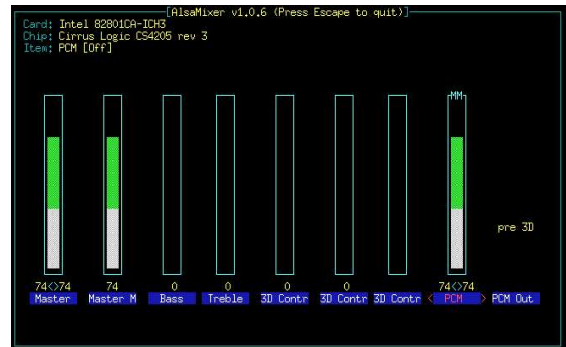
3.1.2 Verschiedene Soundserver

Verschiedene Soundserver können nicht gleichzeitig auf einer Audiotarte laufen, es sei denn, dass das Dmix-Plugin eingerichtet ist. Angenommen, Ihr Medienabspielprogramm basiert ausschließlich auf aRts. Gleichzeitig möchten Sie aber via eines Softwaretelefones oder einer Videokonferenzsoftware erreichbar sein, die ausschließlich via esound Audioausgabe vornehmen können. Dann werden Sie, sobald ein Teilnehmer Sie anrufen möchte, das Klingeln nicht hören können, da aRts das Audiogerät für andere Programme und Soundserver blockiert. In diesem Falle müssen Sie notgedrungen auf eine der beiden Applikationen verzichten oder aber alternative Programme verwenden, die ihre Klänge alle auf den selben Soundserver ausgeben können.⁹

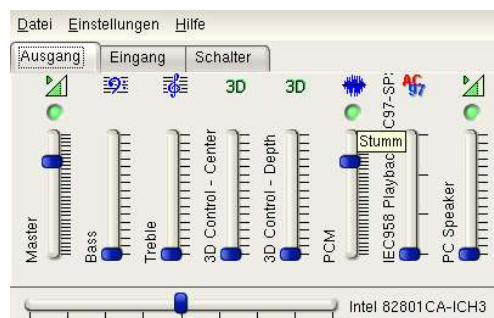
3.1.3 Mixereinstellungen

Wenn Sie mehrere Audioprogramme laufen lassen, die tatsächlich den selben Soundserver verwenden, Sie aber dennoch keinerlei Audioausgabe erhalten, dann sollten die Mixereinstellungen und die zum Verstärker- bzw.

Lautsprechersystem führende Verkabelung geprüft werden. Programme schicken Ihre Audioströme, gegebenenfalls über einen Soundserver, an den PCM-Anschluss der Soundkarte. Daher ist im Mixerprogramm insbesondere sicherzustellen, dass der Masterregler und der PCM-Regler geöffnet sind, und dass der PCM-Kanal nicht stummgeschaltet (engl. Mute) ist. Im nachfolgenden Beispiel sind zwar die benötigten Regler aufgezogen, der PCM-Kanal ist jedoch stummgeschaltet:



Reaktivieren Sie den PCM-Kanal, indem Sie diesen in Alsamixer via der Cursortasten anwählen und dann die Taste M auf der Tastatur drücken, so dass die beiden »M« über dem PCM-Kanal ausgeblendet werden. In kmix finden Sie über dem PCM-Kanal eine LED-Schaltfläche, mittels derer sich Kanäle stummschalten lassen:



3.1.4 Dmix

Wie bereits angesprochen, werden künftige ALSA-Versionen in der Lage sein, via des Dmix-Plugins mehreren Programmen den Zugriff auf eine einzelne Audiotarte zu ermöglichen. Derzeit ist jedoch Dmix noch nicht sehr verbreitet. Eine weite Verbreitung von Dmix ist jedoch sehr wünschenswert und wird aller Voraussicht nach auch stattfinden. Ist Dmix eingerichtet, können Sie

auch mehrere Soundserver wie aRts, esound und JACK gleichzeitig auf einer Audiokarte laufen lassen.

3.2 MIDI-Wiedergabe

Gelegentlich besteht der Wunsch, MIDI-Dateien wiederzugeben. Zuvor sollte verstanden sein, was MIDI eigentlich ist. MIDI beinhaltet – im Gegensatz zu Ogg Vorbis, mp3, flac oder Wave-Audio – selbst keinerlei Informationen über den abzuspielenden Klang, sondern Abspielinformationen ähnlich gedruckter Noten. Um gedruckte Noten wiedergeben zu können, benötigt man das passende Instrument und einen Musiker. Um MIDI-Dateien wiedergeben zu können, benötigen Sie anstatt des Musikers eine MIDI-Abspielsoftware und statt eines echten Instrumentes elektronische oder Softwareinstrumente. An dieser Stelle sei aus Platzgründen auf MIDI nicht weiter eingegangen, da es vornehmlich ein Dateiformat ist, welches in Musikkreisen verbreitet ist und bei Nicht-Musikern kaum verwendet wird.

3.3 Aufzeichnung

Bei der Aufzeichnung stellt sich ebenfalls die Frage, über welches Audiosystem aufgenommen werden soll. Die meisten Aufnahmeprogramme erwarten ein bestimmtes Audiosystem als Voraussetzung. Ist dieses nicht vorhanden, kann das Programm nicht arbeiten.

3.3.1 Direktzugriff auf die Audiokarte

Wie schon bei der Wiedergabe, so kann ein Programm auch bei der Aufzeichnung direkt auf die Audiokarte zugreifen. Dies ist bei qarecord[19] standardmäßig der Fall:



3.3.2 Aufzeichnung über einen Soundserver

Teil des KDE-Projektes ist das Programm Krec. Es benötigt zur Aufnahme den KDE-Soundserver aRts. Selbst jedoch wenn aRts bereits läuft, genügt dies nicht, denn aRts läuft standardmäßig nur im Wiedergabemodus. Man muss daher aRts über das KDE-Kontrollzentrum zuerst in den Vollduplexmodus schalten:

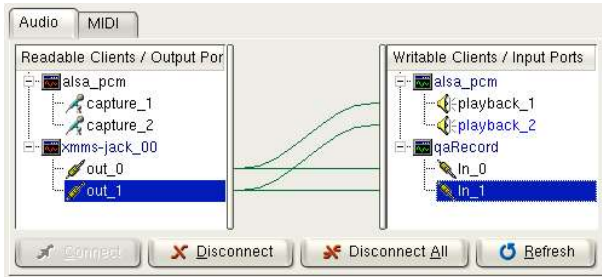


qarecord kann nicht nur über Direktzugriff auf die Audiokarte, sondern mittels einer Kommandozeilenoption beim Start auch über den Soundserver JACK⁸ aufnehmen:



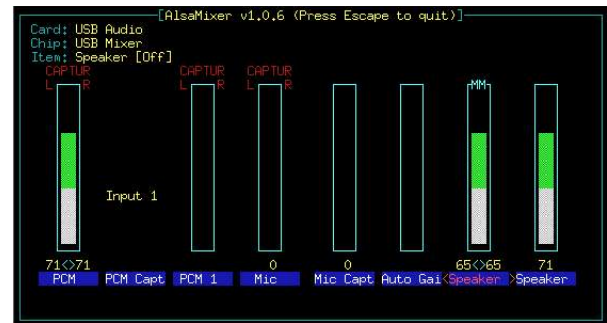
Sehr komfortabel erfolgt die Auswahl der Aufnahmequelle, wenn die Aufnahme via JACK

als Audiosystem läuft. Läuft sowohl JACK als auch qarecord, stellt man über eine der graphischen JACK-Patchbays die Audio-Verbindung zwischen dem aufzunehmenden JACK-Port (beispielsweise die Audiokarte oder ein anderes im JACK-Modus laufendes Programm) und qarecord her. Beispielsweise kann man xmms als JACK-Applikation laufen lassen, einen Radiostream hören und dies gleichzeitig via qarecord auf Festplatte aufzeichnen:



3.3.3 Mixereinstellungen

Bei der Aufzeichnung von einer Audiokarte sind verschiedene Mixereinstellungen zu treffen. Manche Karten haben verschiedene Audioeingänge wie beispielsweise einen Mikrophoneingang, einen Lineeingang und einen optischen Eingang. Daher wählt man zuerst, beispielsweise via alsamixer, den gewünschten Eingang aus, im abgebildeten Beispiel den Input 1 einer USB-Soundkarte. Im zweiten Schritt startet man die Wiedergabe des aufzunehmenden Audiosignales, beispielsweise indem man die Stereoanlage mit der Soundkarte verbindet und regelt dann über den Regler, der dem aufzunehmenden Eingang zugeordnet ist (im abgebildeten Beispiel der Regler ganz links, der mit PCM bezeichnet ist), den Aufnahmepegel so ein, dass während der Aufnahme auch bei lauten Passagen der rote Aussteuerungsbereich in qarecord nicht erreicht wird, da es ansonsten zu unschönen Verzerrungen kommt und die Aufnahme unbrauchbar wird:



3.3.4 Dsnoop

Beim Aufzeichnen gilt das selbe wie bei der Wiedergabe. Man kann lediglich mit einem Programm direkt von der Soundkarte aufnehmen. Unter Zuhilfenahme eines Audioservers kann man theoretisch mit mehreren Applikationen gleichzeitig aufnehmen. Steht kein Audioserver zur Verfügung, kann man – ähnlich dem Dmix-Plugin zur Wiedergabe – das Dsnoop-Plugin verwenden, um mehreren Programmen gleichzeitig die Weiterverarbeitung von Audioströmen einer einzigen Karte zu gestatten. Dies kann beispielsweise dazu verwendet werden, einen Radiostream über den Rechner aufzuzeichnen und gleichzeitig mitzuhören.

3.3.5 Stolperstein

Ein häufiger Stolperstein bei der Aufnahme ist der Direktdurchschleifweg der Audiokarte. Über den Direktdurchschleifweg kann man Audiodaten, die an der Karte ankommen, direkt auf den Kartenausgang wieder ausgeben, so dass man das ankommende Signal direkt hören kann. Diese Audiodaten haben dabei aber die Soundkarte noch gar nicht in Richtung Hauptplatine des Rechners verlassen, sondern werden nur innerhalb der Karte gleich wieder an den Ausgang durchgereicht. Eine häufig gestellte Frage lautet daher immer wieder, dass man zwar das aufzunehmende Signal durch die Lautsprecher hören kann, aber nichts beim Aufnahmeprogramm ankommt. In diesem Falle ist zuerst zu prüfen, ob im Mixerprogramm der richtige Audiokarteneingang zur Aufnahme eingestellt ist, dass dessen Aufnahmekanal nicht stummgeschaltet ist und in der Lautstärke korrekt eingeregelt wurde. Jetzt ist noch sicherzustellen, dass das Aufnahmeprogramm über das Audiosystem, das es nutzt, an die eingehenden Audiodaten herankommt.

3.4 MIDI-Aufzeichnung

MIDI ist, wie bereits angesprochen, ein Dateiformat für Musiker. Musiker zeichnen Ihr Spiel via spezieller Soundkarten mit MIDI-Schnittstellen, die beispielsweise an den USB-Bus angeschlossen sind, auf. Nichtmusiker werden nicht in die Verlegenheit kommen, MIDI aufzuzeichnen, da man beispielsweise Radiosendungen oder Schallplatten nicht als MIDI aufzeichnen kann.

4 Zusammenfassung & Ausblick

Audio auf einem linuxbasierten Rechner zu verwenden, ist auch im Jahre 2005 noch nicht ganz trivial für einen Durchschnittsanwender. Mit etwas Hintergrundwissen lassen sich jedoch alle Aufgaben durchaus erledigen – gelegentlich sogar komfortabler als unter anderen Betriebssystemen.

Die nächsten zwei bis drei Jahre werden weitere Vereinfachungen mit sich bringen, einerseits durch weitere Verbreitung des ALSA-Plugins Dmix, andererseits dadurch, dass sich voraussichtlich das KDE- und das Gnome-Projekt auf einen gemeinsamen Soundserver einigen werden. Eine Beispielkonfiguration könnte beispielsweise so aussehen, dass auf dem Rechner ALSA mit dem Dmix-Plugin läuft. Darauf setzt dann Gstreamer und parallel dazu JACK auf. Audioprogramme, die direkten Zugriff auf das Audiogerät benötigen, können diesen erhalten, ohne das Audiogerät für andere Programme zu blockieren. Die beiden Soundserver Gstreamer und JACK können gleichzeitig laufen und auch Audio ausgeben:

KDE- und Gnome-Programme	Professionelle Musikprogramme	Programme mit Audiokarten-direktzugriff
Gstreamer	JACK	
ALSA mit Dmix und Dsnoop-Plugins		
Kernel		
Hardware		

Insofern dürfen wir auf die Entwicklungen in naher Zukunft gespannt sein. Dmix wird vieles einfacher und transparenter machen, nicht nur für den Anwender, sondern auch für die Entwickler.

5 Fußnoten

¹ Treiber heißen unter Linux normalerweise »Module«; in diesem Dokument werden die Module jedoch ab sofort als »Treiber« bezeichnet, um Umsteigern das Verständnis zu erleichtern

² Um genau zu sein, wird dabei die ALSA-Schnittstelle `alsa-lib` aktiviert; die eigentlichen Treiber stünden auch ohne ALSA-Start zur Verfügung, könnten jedoch nicht bzw. nur umständlich angesprochen werden

³ Die Bezeichnung »Mixer« ist irreführend, denn das eigentliche Mixen übernimmt die Audiokarte; der »Mixer« ist also lediglich eine Steuerung für das Verhalten der Soundkarte

⁴ Es gibt Audiokarten für den professionellen Einsatz, die via Hardwaremixing in der Lage sind, mehrere Datenströme entgegenzunehmen und zu mischen

⁵ `aRts` ist in der Lage, das Audiogerät nach einer bestimmten Zeitspanne freizugeben, wenn gerade keine Audioausgabe erfolgt; daher kann es vorkommen, dass ein Programm auch bei laufendem `aRts` Zugriff auf die Audiokarte erhält

⁶ Es gibt auch KDE-Programme, die nicht zwingend auf `aRts` angewiesen sind

⁷ Für die systemweite Konfiguration legt man die Datei `/etc/asoundrc` an, für eine benutzerspezifische Konfiguration `~/.asoundrc`

⁸ Der Nachteil an `qarecord` besteht darin, dass man es sich selbst aus dem Web besorgen und kompilieren muss, wobei Verzeichnispfade im `Makefile` auf anderen Distributionen als SuSE anzupassen sind – für Durchschnittsanwender nicht ganz trivial

⁹ Eine weitere Möglichkeit besteht darin, einen Soundserver statt auf die Audiokarte auf einen anderen Soundserver ausgeben zu lassen; so kann beispielsweise `aRts` direkt in `JACK` ausgeben

6 Quellen

- [1] <http://www.opensound.com/linux.html>
- [2] <http://www.alsa-project.org/>
- [3] <http://www.novell.com/de-de/linux/suse/>
- [4] <http://www.mandriva.com/>
- [5] <http://www.debian.de/>
- [6] <http://www.gentoo.org/>
- [7] <http://www.kde.de/>
- [8] <http://www.gnome.de/>
- [9] <http://0pointer.de/lennart/projects/polypaudio/>
- [10] <http://jackit.sourceforge.net/>
- [11] <http://muse.serverkommune.de/>
- [12] <http://qjackctl.sourceforge.net/>
- [13] <http://www.scs.carleton.ca/~drobilla/patchage/>
- [14] <http://alsa.opensrc.org/index.php?page=AlsaSharing>
- [15] <http://www.mediaapplicationserver.net/>
- [16] <http://gstreamer.freedesktop.org/>
- [17] <http://www.portaudio.com/>
- [18] <http://audacity.sourceforge.net/>
- [19] <http://alsamodular.sourceforge.net>

7 Lizenz

Der Urheber dieses Dokumentes ist Christoph Eckert. Es wurde im April 2005 zu den Bedingungen der "GNU free documentation license (GFDL)" veröffentlicht.