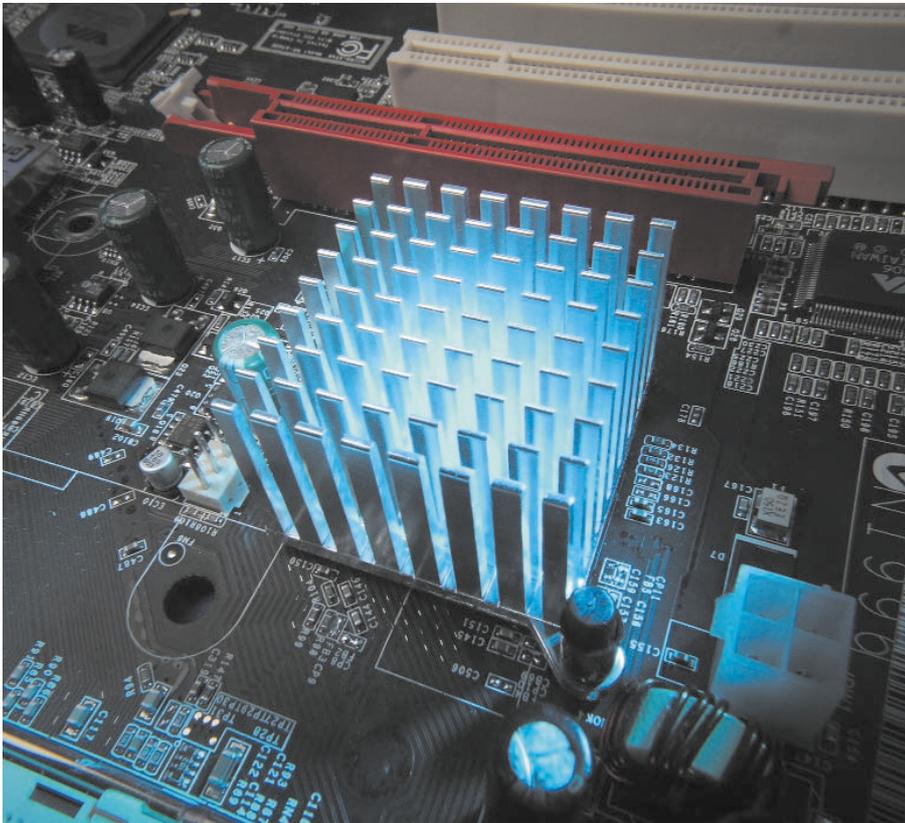


## 64-BIT-LINUX IN DER PRAXIS

# BITTE 64 BIT!

AMDs 64-Bit-Erweiterung ist seit über drei Jahren auf dem Markt, und Intel hat jüngst nachgezogen. Trotz signifikantem Marktanteil der Prozessoren kommen 64-Bit-Betriebssysteme auf dem Desktop nur langsam in Fahrt. Woran liegt es?

VON **MATTIAS SCHLENKER**



**K**napp alle zehn Jahre steht der Schrecken von Programmierern und Hardware-Herstellern an: der Architekturwechsel. Größere Festplatten und Arbeitsspeicher verlangen nach einem größeren Adressraum, und gestiegene Anforderungen an die Genauigkeit von Berechnungen erfordern die Verarbeitung langer Fließkommazahlen in einem Happen. Viele Prozessorhersteller versuchen, Architekturwechsel so sanft wie möglich zu gestalten. Sie versuchen, die neuen Prozessoren abwärtskompatibel zu gestalten, was Anwendern und Software-Herstellern gleichermaßen einen sanften Umstieg garantieren soll. Seit Anfang der 1980er garantierte Intel Abwärtskompatibilität, doch mit dem Umstieg

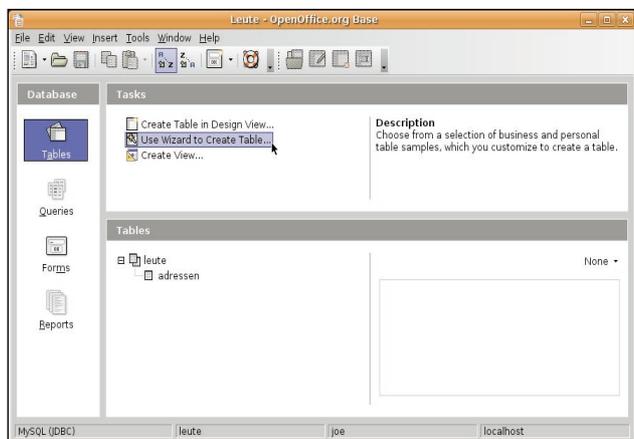
auf 64 Bit wollte Intel Altlasten über Bord werfen. Intels Itanium genannte Architektur IA64 konnte 32-Bit-Code nur in einer sehr langsamen Emulation ausführen. Die neuen Prozessoren waren ab 2005 nur für Server und Workstations erhältlich und sollten

nach einigen Jahren auch den Desktop erobern. Doch Intel hatte die Rechnung ohne den Erzrivalen AMD gemacht. AMD erweiterte den Befehlssatz der Pentium-Prozessoren und deren Register so, dass ein Pentium-kompatibler 64-Bit-Prozessor entstand. Diese Prozessoren wurden ab 2003 in großen Stückzahlen unter das Volk gebracht und umgingen das Henne-Ei-Problem eines radikalen Architekturwechsels. Tatsächlich fand der AMD64 gerade bei Linux-Anwendern regen Zuspruch: Schließlich konnten Anwender im wissenschaftlichen Bereich genauso wie die Betreiber von Servern tatsächlich schnell von den Möglichkeiten wie der einfachen Adressierung von mehr als 4 Gigabyte Speicher profitieren, für die sie bislang auf teure RISC-Rechner beispielsweise von Sun oder Alpha zurückgreifen mussten.

Auf AMD Athlon 64 und Intel EMT 64 (die meisten Core-Duo- und Core-2-Duo-Prozessoren sowie neuere Pentium-4-Modelle) abgestimmte Pakete bieten alle Distributoren. Wo nur freie Software zum Einsatz kommt, war die Anpassung der Software auf 64 Bit kein großes Problem und teilweise schon lange vor der Jahrtausendwende fertiggestellt: Viel Vorarbeit für den aktuellen Architekturwechsel leisteten die Programmierer, die Linux ab 1994 an den damals bereits 64-bittigen Alpha-Prozessor anpassten. Zudem waren ab 1995 64-bittige SPARC-CPU's erhältlich, die wie die Al-

```
mattias@amd64-test: ~
mattias@amd64-test:~$ which firefox
/usr/bin/firefox
mattias@amd64-test:~$ file /usr/bin/firefox
/usr/bin/firefox: symbolic link to `../lib/firefox/firefox'
mattias@amd64-test:~$ file /usr/bin/../lib/firefox/firefox
/usr/bin/../lib/firefox/firefox: Bourne shell script text executable
mattias@amd64-test:~$ file /usr/lib/mozilla-firefox/firefox-bin
/usr/lib/mozilla-firefox/firefox-bin: ELF 64-bit LSB executable, AMD x86-64, version 1 (SYSV), for GNU/Linux 2.6.0, dynamically linked (uses shared libs), for GNU/Linux 2.6.0, stripped
mattias@amd64-test:~$ file /usr/lib/openoffice/program/soffice.bin
/usr/lib/openoffice/program/soffice.bin: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), for GNU/Linux 2.2.0, dynamically linked (uses shared libs), for GNU/Linux 2.2.0, stripped
mattias@amd64-test:~$
```

**Die Tools *file* und *which* helfen beim Aufspüren von 32-Bit-Binaries. In vielen Fällen ist aber ein Blick in Wrapperscripte erforderlich.**



**OpenOffice.org unter Ubuntu: Erst im 32-Bit-Chroot konnten wir erfolgreich mit dem JDBC-Datenbanktreiber arbeiten.**

phas gerne als Webserver und Workstation genutzt wurden, sodass Apache, Perl, PHP und MySQL Ende der 1990er und damit lange vor AMD64/EM64T bereit für 64 Bit waren und mit der Verfügbarkeit der 64 Bit-Prozessoren im Consumer-Bereich sofort eingesetzt werden konnten.

Etwas schleppender ging die Einführung von 64-bittiger Software im Desktop- und Workstation-Bereich vorwärts. Wir haben die breite Verfügbarkeit von Intels EMT64-Prozessoren (ein Technologie-Tauschabkommen erlaubt es Intel, AMDs Erweiterungen in eigene Prozessoren zu integrieren) zum Anlass genommen, den Stand der Dinge auf dem Linux-Desktop zu untersuchen.

### Mit Speed

Zweifelsfrei kann der 64-Bit-Prozessor dort gegen 32 Bit punkten, wo die parallele Verarbeitung der Datenmenge ins Gewicht fällt. Als kleinen Benchmark haben wir dafür eine WAV-Datei nach MP3 encodiert, ein JPEG-Foto gedreht und den Linux-Kernel entpackt. Um nicht versehentlich die Geschwindigkeit der Festplatte zu testen, haben wir die Ursprungsdatei einmal nach `/dev/null` umgeleitet und so in den Dateisystem-Cache gepackt. Die Ausgabe wurde nicht auf Platte geschrieben, sondern ebenfalls ins Datengrab `/dev/null` geschickt.

Testrechner war ein Sockel-939-Athlon 64 3000+ (Single Core) mit 2 GByte RAM. Beide Tests wurden unter dem 64-Bit-Kernel durchgeführt und die Tests mit dem 32-Bit-Userland nochmals unter einem 32-Bit-Kernel verifiziert.

Beim Entpacken des Linux-Kernels benötigte das 32-Bit `bunzip2` mit 23 Sekunden fast 30% mehr Zeit als das 64-Bit-Pendant, welches nach 18 Sekunden fertig war. Auch beim Test mit der Stapelbildverarbeitung `convert` (aus dem Paket `Image Magick`)

konnte das 64-Bit-Binary der 32-Bit-Version mit 4,6 statt 7,8 Sekunden für das Drehen einer 5 Megapixel großen JPEG-Datei um 15 Grad davonziehen.

Einzig beim Enkodieren von „Dream Theaters“ Dreizehn-Minuten-Stück „Trial of Tears“ von WAV nach MP3 mit 192kBit/s gewann das 32-Bit-Binary knapp: 63 Sekunden brauchte der 32-Bit-LAME, das 64-Bit-Gegenstück war zwei Sekunden langsamer.

### Kompatibilitätsfragen

Anders als auf dem Server, wo Hersteller von Closed Source Software längst 64-bittige CPUs ausnutzen, wenn sie bereitstehen, sieht es auf dem Desktop aus: Der *Adobe Acrobat Reader*, das *Flash-Plug-in* oder Suns *Java-Plug-in* für Firefox sind nur als 32-Bit-Versionen verfügbar, und bei *OpenOffice.org* ist die Portierung nach 64 Bit noch nicht abgeschlossen. Als Standalone-Anwendungen sollten diese – zumindest theoretisch keine Probleme bereiten – schließlich kann der 64-Bit-Kernel auch 32-Bit-Anwendungen ausführen.

Unmöglich ist jedoch die Verwendung von 32-Bit-Plug-ins im 64-Bit-Browser. Strategien, dieses Dilemma zu lösen, gibt es viele – und sie wurden von den Distributoren unterschiedlich gut umgesetzt.

### Ubuntu 6-06 LTS („Dapper Drake“)

Ubuntu bringt *Firefox* als 64-Bit-Version mit. Sehr konsequent – nur leider unterstützt diese keine 32 Bit-Plug-ins. *Java* und *Flash* müssen deshalb außen vor bleiben. Selbst bei Solaris 10 auf der exotischen Sparc64-Plattform hat den Nutzer die Möglichkeit, *Java-Plug-ins* zu nutzen.

Kaum besser ist die Situation bei *OpenOffice.org*: Dass *OpenOffice.org* in der 64-Bit-Version bislang nicht über das Beta-Stadi-

um hinausgekommen ist, darf als Folge einer fünfzehn Jahre alten Codebasis enormen Umfangs gesehen werden. Da *OpenOffice*-Prozesse selbst selten Größen im GByte-Bereich annehmen, ist die Arbeit mit der 32-Bit-Version zu verschmerzen.

Problematisch ist allerdings häufig die Einbindung der 64 Bit-Java-Umgebung. Wer intensiv mit Datenbanken arbeitet und auf den JDBC-Treiber angewiesen ist, stößt deshalb bei Ubuntu möglicherweise an die Grenzen der installierten freien GCJ-Umgebung.

Adobes *Acrobat Reader* war auch im Multi-verse-Repository nicht verfügbar. Wir installierten den PDF-Betrachter deshalb aus dem von Adobe bereitgestellten `.tar.gz`-Ar-

## Stolpersteine

➤ **Ndiswrapper:** Das Kernelmodul, welches die Verwendung von Windows WLAN-Treibern unter Linux erlaubt, existiert zwar mittlerweile in einer 64-Bit-Version, die aber nur Windows 64-Bit-Treiber laden kann. Die Auswahl kompatibler Treiber ist jedoch derzeit sehr übersichtlich. Wenn Sie auf *ndiswrapper* angewiesen sind, ist in den meisten Fällen der Griff zur 32-Bit-Distribution empfehlenswert. Gleiches gilt für den *Driverloader* von *Linuxant* und den Nutzern von *FreeBSD* und *OpenSolaris*, wo ein mit *ndiswrapper* verwandtes Treiberkonzept erhältlich ist.

**Qemu:** Die freie VMware-Alternative harmoniert gut mit 64-Bit-Prozessoren. Um Gebrauch vom Beschleunigermodul machen zu können, sollten Sie jedoch mit dem Befehl `qemu-system-x86_64` für den Start virtueller 64-Bit-Rechner verwenden.

**AVM Fritz!Card:** Stichproben auf der Webseite von AVM ergaben eine breite Verfügbarkeit von 64-Bit-Treibern, die teils ein neueres Build-Datum aufwiesen als die 32-Bit-Gegenstücke. Lediglich kleine Helferlein für die Überwachung der Leitungsaktivität stellt AVM nur in 32-Bit-Versionen bereit.

**Mplayer:** Das Video-Abspielprogramm *Mplayer* entfaltet seinen gesamten Funktionsumfang erst wenn es Windows-Codecs laden kann. Die 64 Bit-Variante kommt jedoch mit 32 Bit-Codecs nicht klar, weshalb gegebenenfalls ein 32 bittiger *Mplayer* in einer Chroot-Umgebung installiert werden sollte.

chiv. Bis auf kleinere Darstellungsfehler der Bedienelemente funktionierte der Reader einwandfrei – offenbar genügen die von *OpenOffice* mitinstallierten GTK-Kompatibilitätsbibliotheken.

### SUSE Linux / openSUSE 10.1

Deutlich besser sah die Situation bei openSUSE aus: *Firefox* liegt als 32-bitiges Binary bei, das 32-Bit-Plug-ins verträgt: Flash funktionierte im Browser auf Anhieb.

Wer *Java* mit *Firefox* und *OpenOffice.org* nutzen möchte, sollte die 32-Bit-Version (1.4.2) wählen.

Die dürfte allerdings Java-Entwickler nicht immer zufrieden stellen. Immerhin lassen sich beide Versionen parallel installieren und betreiben, gegebenenfalls ist beim Aufruf der 64-Bit-Version (Java 1.5.0) eine Anpassung der Umgebungsvariablen *PATH* und *CLASSPATH* erforderlich. Da SUSE eine große Auswahl an 32-Bit-Kompatibilitätsbi-

bliotheken bereitstellt, verläuft auch die Installation und anschließende Nutzung vom *Adobe Acrobat Reader* oder *RealPlayer* problemlos.

Der einzige Vorwurf, den man SUSE noch machen kann, ist, dass einige der Kompatibilitätsbibliotheken nutzlos auf der Platte herumlungern dürften. Doch besser ein paar MByte Plattenplatz verschenkt, als Startschwierigkeiten bei manuell nachinstallierten 32-Bit-Applikationen.

## 32-Bit-Chroot-Umgebung unter Ubuntu und Debian

Da die Unterstützung von 32-Bit-Applikationen unter Ubuntu durchwachsen und unter Debian nicht vorhanden ist, ist eine *chroot*-Umgebung der sauberste Weg, Altlasten ausführen zu können. Im Prinzip können Sie diese Anleitung auch auf SUSE oder Fedora übertragen (wenn Sie dort eine vollständige *chroot*-Umgebung benötigen), allerdings verwenden diese Distributionen eine andere Taktik für die Installation.

Für den Aufbau der *chroot*-Umgebung benötigen Sie zwei Tools: *debootstrap* hilft bei der Installation eines minimalen Debian- oder Ubuntu-Systems in einem Verzeichnis, während *dchroot* auch normalen Nutzern den Weg in die *chroot*-Umgebung öffnet: Das Kommando *chroot* ist Administratoren vorbehalten.

Sind beide Programme mit *apt-get* installiert, kann die Bevölkerung des *chroot* beginnen:

● Erstellen Sie ein Verzeichnis für den *chroot*-Käfig und installieren Sie dort mittels *debootstrap* Ubuntu:

```
mkdir -p /usr/local/chroot32
debootstrap --arch i386 dapper \
/usr/local/chroot32 \
http://de.archive.ubuntu.com/ubuntu
```

Wenn Sie Debian als Basis verwenden, ist der Code-Name der Distribution und der Server-Pfad natürlich entsprechend anzupassen. In der Regel werden Sie *etch* oder *sid* verwenden wollen.

● Im *chroot*-Käfig benötigen Sie eine */etc/apt/sources.list*. In der Regel genügt es, die des Host-Systems zu kopieren:

```
cp /etc/apt/sources.list /usr/local/chroot32/etc/apt/
```

● Einige Verzeichnisse des Host-Systems müssen Sie ins *Chroot*-System einhängen. Die Mount-Option *bind* sorgt dafür, dass ein Verzeichnis an einem anderen Ort des Dateisystems erscheint. Wir fügten folgende Zeilen an die */etc/fstab* an, um dies zu bewerkstelligen:

```
/home /usr/local/chroot32/home none bind 0 0
/tmp /usr/local/chroot32/tmp none bind 0 0
```

Gleiches gilt für die „speziellen“ Dateisysteme *devpts* und *proc*:

```
devpts /usr/local/chroot32/dev/pts devpts none 0 0
proc /usr/local/chroot32/proc proc none 0 0
```

● Auch einige Konfigurationsdateien müssen im Jail vorhanden sein. Sie können diese per Hardlinks verfügbar machen (Softlinks funktionieren nicht!), sie mit *rsync* kopieren oder wie wir mit der Mount-Option *bind* in die */etc/fstab* eintragen:

```
/etc/passwd /usr/local/chroot32/etc/passwd none bind 0 0
/etc/shadow /usr/local/chroot32/etc/shadow none bind 0 0
/etc/group /usr/local/chroot32/etc/group none bind 0 0
/etc/sudoers /usr/local/chroot32/etc/sudoers none bind 0 0
```

```
/etc/hosts /usr/local/chroot32/etc/hosts none bind 0 0
/etc/resolv.conf /usr/local/chroot32/etc/resolv.conf none
➔ bind 0 0
/etc/nsswitch.conf /usr/local/chroot32/etc/nsswitch.conf
➔ none bind 0 0
```

● Dateien, die mit *bind* gemountet werden, müssen existieren. *touch* legt sie als leere Dateien an, sofern sie bislang fehlen, aktualisiert bei vorhandenen Dateien aber lediglich den Zeitstempel:

```
touch /usr/local/chroot32/etc/passwd
touch /usr/local/chroot32/etc/shadow
touch /usr/local/chroot32/etc/group
touch /usr/local/chroot32/etc/sudoers
touch /usr/local/chroot32/etc/hosts
touch /usr/local/chroot32/etc/resolv.conf
touch /usr/local/chroot32/etc/nsswitch.conf
```

● Lassen Sie die noch nicht eingebundenen Dateien und Verzeichnisse mounten:

```
mount -a
```

● Wechseln Sie als Root in den Käfig und aktualisieren Sie dort zunächst die installierte Software, installieren Sie gegebenenfalls benötigte Programme nach:

```
chroot /usr/local/chroot32 /bin/bash -l
apt-get update
apt-get upgrade
apt-get install firefox
apt-get install mozilla-firefox-locale-de-de
```

Auch das Flash-Plug-in können Sie nun installieren:

```
apt-get install flashplugin-nonfree
```

● Damit der *Chroot*-Käfig von normalen Nutzern betreten werden kann, muss die Datei */etc/dchroot.conf* angepasst werden:

```
chroot32 /usr/local/chroot32
```

● Beim ersten Start eines Programmes im *chroot* als normaler User sollten Sie die Umgebungsvariable für das *DISPLAY* übergeben:

```
dchroot DISPLAY=$DISPLAY firefox
```

● Um nicht bei jedem Aufruf eine ellenlange Kommandozeile eingeben zu müssen, erstellten wir ein kleines ausführbares Shellscript */usr/bin/firefox32*, das Umgebungsvariablen und Kommandozeilen-Parameter übergibt:

```
#!/bin/bash
dchroot \
DISPLAY=$DISPLAY \
LANG=$LANG firefox $*
```

Das war es! Sie können nun entscheiden, wann Sie den etwas flotteren 64-Bit-Firefox nutzen wollen oder die mit *Flash* und *Java* ausgestattete 32-Bit-Version. Lesezeichen und Einstellungen werden von beiden geteilt. Verwenden Sie die Versionen deshalb nie gleichzeitig.



Der im *chroot* installierte 32-Bit-Firefox kann das Flash-Plug-in verwenden. Ungewohnt: Hier ist noch nicht das Ubuntu-Thema installiert.

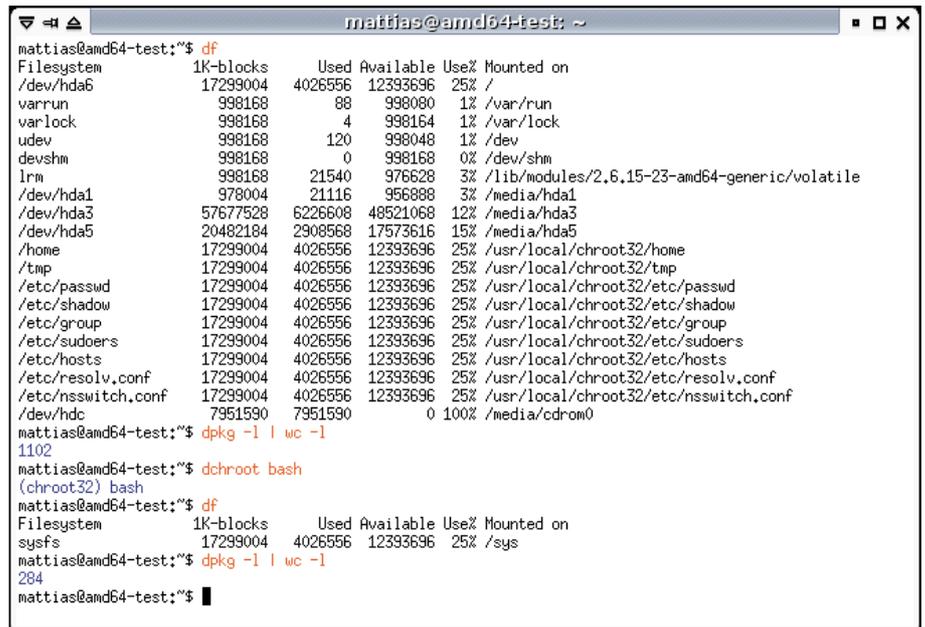
### Debian / Kanotix

Debian und das auf der „instabilen“ Version dieser Distribution basierende Kanotix verzichten als einzige Distributionen ganz auf die Mischung von 32-Bit- und 64-Bit-Komponenten.

In der Folge kann man unter einem frisch installierten Debian weder Browser-Plugins noch *OpenOffice.org* benutzen. Anstatt 32-Bit-Bibliotheken und Applikationen bunt ins 64-Bit-System einzustreuen, empfehlen die meisten Debian-Kenner den Aufbau eines *chroot*-Containers, der den 32-Bit-Anwendungen eine Heimat gibt.

Ein *chroot*-Container – manchmal auch Jail oder Käfig genannt – ist nichts anderes als eine komplette Installation eines Betriebssystems in einem Verzeichnis, beispielsweise */usr/local/chroot32*. Mit dem Systemaufruf *chroot* wird dieses Verzeichnis für einen Prozess (und dessen Kindprozesse) zum Wurzelverzeichnis. Bereits Ende der 1970er wurde *chroot* eingeführt, um Bibliotheken und Programme eines frisch kompilierten Systems testen zu können, ohne neu starten zu müssen. Seither ist *chroot* für die verschiedensten Einsatzbereiche sehr beliebt: sei es, um Server-Dienste in einer ausbruchsicheren Umgebung einzusperren, das Dateisystem von Live-CDs zu verändern oder ein komplettes 32-Bit-System innerhalb eines 64-Bit-Systems zu betreiben. Eine so klare Abgrenzung ist unserer Ansicht nach die beste

Möglichkeit, 32-Bit-Altlasten ins 64-Bit-System zu integrieren. Ein so komplett vorgehaltenes *chroot*-System hat freilich auch seine Nachteile: Es ist deutlich umfangrei-



Mit *dchroot* kann auch eine Shell im *chroot* geöffnet werden. Im Zweifel zeigt *df*, wo Sie sich gerade befinden.

cher als einzelne Kompatibilitätsbibliotheken und muss wie ein eigenes System regelmäßig aktualisiert werden.

### Fazit

Die Programmierer freier Software haben ihre Hausaufgaben erstaunlich gut gemacht

und die über drei Jahre währende Verfügbarkeit der Prozessoren dazu genutzt, fast alle im Quellcode erhältlichen Anpassungen fit für 64 Bit zu machen.

Absolut unbefriedigend sind die Fortschritte bei unfreier Software wie dem *Flash-Plug-in* oder *Adobes Acrobat Reader*.

Auch Sun hat die 64-Bit-Version seines Javas eher für Server und Entwickler-Workstations als den normalen Desktop-Nutzer vorgesehen.

Den Distributoren kommt die undankbare Aufgabe zu, Altes und Neues zu einem sauberen Ganzen zu kombinieren.

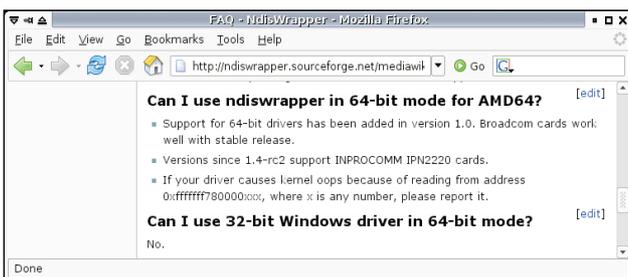
Entscheiden sie sich für den 32-Bit-Firefox, um kommerzielle Plug-ins zu ermöglichen, setzen sie sich dem Verdacht aus, die Wei-

terentwicklung des Browsers zu sabotieren, bieten sie stattdessen die 64-Bit-Version an, muss der Nutzer auf Plug-ins verzichten.

Eine klare Empfehlung für oder wider 64 Bit können wir nicht abgeben: Wenn Sie bei Einzelanwendungen wie der Video-Transcodierung von 64 Bit profitieren können, sollten Sie auf jeden Fall zur AMD-/EMT64-Version Ihrer Distribution greifen und gegebenenfalls 32-Bit-Anwendungen in den Chroot-Käfig sperren.

Dient Ihr Rechner eher als Büro-PC oder Desktop-Surfmaschine ohne Fokus auf Höchstleistung, ist es derzeit auf jeden Fall pragmatischer beim 32-Bit-System zu bleiben.

jkn



Klare Ansage: Wer 32-Bit-Treiber auf Kernel-Ebene benötigt – egal ob mit *Ndiswrapper* oder nativ – muss das gesamte System im 32-Bit-Modus fahren.