

## VIRTUALISIERUNGSSOFTWARE VIRTUALBOX

# Rechner im Rechner mit VirtualBox

Fehlt Windows-Software unter Linux oder soll unter Windows ein kompletter Linux-Webserver nachgebildet werden, hilft Virtualisierungsssoftware. Eine pfiffige, für Privatanwender kostenlose Lösung verspricht das schwäbische Unternehmen Innotek.

VON **MATTIAS SCHLENKER**



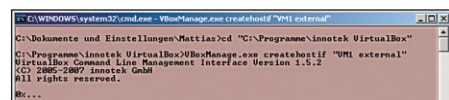
Der Boom der Virtualisierungsprogramme begann Ende der 1990er mit der Software eines kleinen, universitätsnahen Startups: Den Machern von *VMware* gelang es erstmals, auch auf PC-Systemen Zugriffe an den Prozessor des Gastsystems direkt und ohne aufwändige Emulation durchzureichen. In den folgenden Jahren traten weitere Hersteller auf den Markt, die das Thema Virtualisierung mit unterschiedlichen Schwerpunkten aufgriffen. Relativ neu am Markt ist das Weinstädter Unternehmen *Innotek*, das sich einst einen Namen in der OS/2-Welt machen konnte, mit seiner Software *VirtualBox*. Innotek wirbt vor allem mit der Quelloffenheit der Virtualisierungssoftware, welche das Tool auch für den Einsatz in sicherheitskritischen Bereichen qualifiziert: Tatsächlich sind

bei der direkten Konkurrenz *VMware* nur die Kernelmodule offengelegt, beim einzigen komplett freien Virtualisierer *Qemu* (auch von *Xen* und *KVM* herangezogen) fehlt es an einer vergleichbaren Konfigurationsoberfläche. Allerdings sind nicht alle Komponenten von *VirtualBox* quelloffen: Einige Schmankerl wie *iSCSI* oder die Nutzung von USB-Geräten beim Fernzugriff sind der kommerziellen Version vorbehalten.

## Einfache Installation

Die Installation von *VirtualBox* gestaltete sich verhältnismäßig einfach: Innotek bietet neben den Windows- und OSX-Versionen für praktisch jede gängige Linux-Distribution Pakete im jeweiligen Format und deckt neben typischen Enterprise-Distributionen auch ak-

tuelle Consumer- und Community-Versionen ab. Besonders einfach haben es neben den Windows-Nutzern die Nutzer Debian-basierter Distributionen: Bei diesen genügt es, Innoteks Paketschlüssel zu importieren, eine Zeile an die `/etc/apt/sources.list` anzuhängen und die Software mit `apt-get` oder `Synaptic` zu installieren. *VirtualBox* erzeugt bei der Installation eine Gruppe `virtualbox` – nur Benutzer, die Mitglied dieser Gruppe sind, dürfen die Virtualisierungssoftware nutzen. Beachten Sie, dass nach der Änderung



Mit dem Tool *VBoxManage* entsteht das **Bridge-Interface** – auch unter Windows.

der Gruppenzugehörigkeit ein erneutes Login erforderlich ist. Nun lässt sich die Virtualisierungssoftware mit dem Kommando

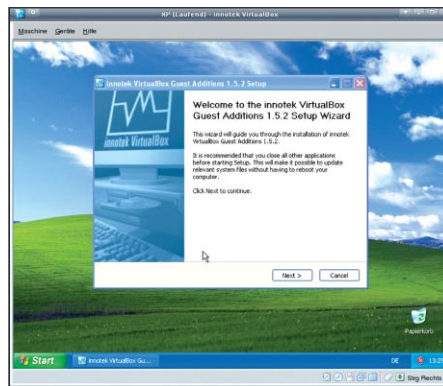
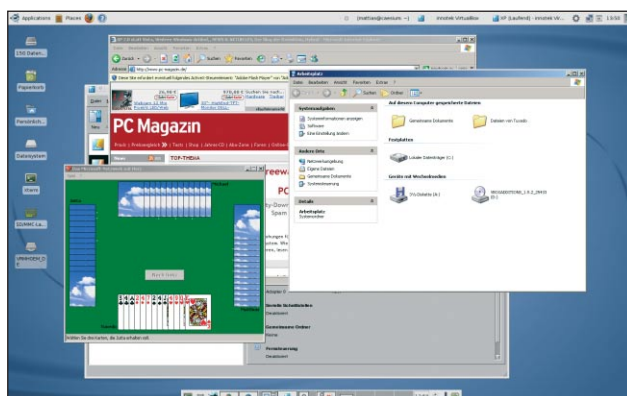
VirtualBox

starten. VMware gibt sich hinsichtlich der Rechtevergabe etwas nutzerfreundlicher und erlaubt allen lokalen Benutzern die Verwendung der Virtualisierungslösung, was aus nachvollziehbaren Gründen in vielen Firmenumgebungen nicht erwünscht ist.

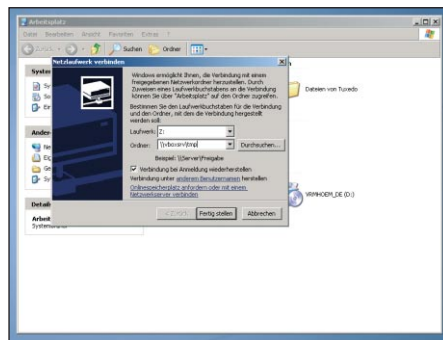
Während VMware bei der Installation relativ tief ins System eingreift und eigene Kernelmodule für den Netzwerkzugriff installiert, geht VirtualBox hier deutlich behutsamer zu Werke: Unmittelbar nach der Installation funktioniert nur die Verbindung nach außen via *Network Address Translation*. Das genügt zum Surfen im Netz und für das Einspielen von Sicherheitsupdates. Weil es für das Wirtssystem und dessen Umgebung jedoch so aussieht, als würde der VirtualBox-Prozess die Verbindungen veranlassen und kein virtuelles Netzwerkinterface sichtbar ist, schlägt der Netzwerkzugriff auf die virtuelle Maschine fehl. Wer VirtualBox dazu nutzen möchte, Serverapplikationen zu testen, muss deshalb den etwas erhöhten Konfigurationsaufwand auf sich nehmen und Netzwerk via Bridging oder Tunneling konfigurieren. Wir ersparten uns zunächst diesen Schritt und legten mit einem VMware-Image von openSUSE 10.3 los, das als VMDK-Datei vorlag.

Gleich beim Start fiel auf, dass VirtualBox nicht ganz so viele VESA-Modi unterstützt wie VMware, der Start bleibt zunächst stehen und erfordert die Drücken der Leertaste. Da VirtualBox zudem einen anderen virtuellen IDE-Controller verwendet, schlug die Suche nach der Rootpartition fehl. Erst nachdem wir die Bootparameter angepasst hatten, funktionierte der Start. Dass keine grafische Oberfläche startete, erwarteten wir angesichts der

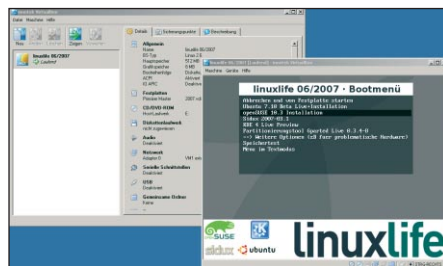
**Killer-Feature:** Fenster des Windows-Gastes erscheinen mit der Tastenkombination *Strg-rechts – L* fast nahtlos auf dem Linux-Desktop, die Windows-Taskleiste wurde hier ausgeblendet.



**Windows als Gast:** Die Installation der „Guest Additions“ startet automatisch vom mitgelieferten ISO-Image.



**Sind die Guest-Additions installiert, können Verzeichnisfreigaben des Wirtes mit geringem Overhead gemountet werden - ein Kernelmodul existiert auch für Linux-Gäste.**



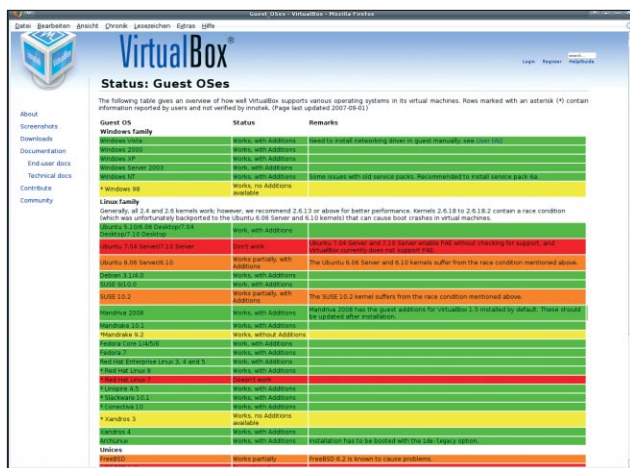
**Windows als Wirt:** Mit VirtualBox und direktem Zugriff auf ein lokales DVD-Laufwerk können Sie die Distributionen der Heft-DVD gefahrlos testen.

Unterschiede der virtuellen Hardware, die Nachkonfiguration mit „sax2“ löste das Problem. Gleiches galt für die virtuelle Netzwerkkarte: Statt Intel E1000 verwendet VirtualBox die virtuelle AMD Lance, welche auch bei Qemu und anderen freien Virtualisierungsprogrammen zum Einsatz kommt. Die Anpassung dauerte gefühlte zehn Minuten und zwei Reboots – durchaus praktikabel, wenn vorhandene Images für die künftige Nutzung unter VirtualBox konvertiert werden müssen, nicht sinnvoll, wenn ein Image ständig zwischen einem VirtualBox- und einem VMware-Nutzer ausgetauscht werden soll.

### VirtualBox nahtlos

Einfacher als die Konvertierung eines fertigen Images ist die „frische“ Installation eines Betriebssystems. Wir erstellten zuerst mit *dd* das ISO-Image einer Windows-XP-Installations-CD und nutzten dann den komfortablen Wizard, um eine Konfiguration zu erzeugen. Die anschließende Installation verläuft wenig spektakulär, wie bei der Konkurrenz fehlen Windows zunächst die Treiber für die virtuelle Netzwerkkarte und Sonderfunktionen der nachgebildeten Grafikkarte. Ein Klick auf *Geräte/Gästeerweiterungen installieren* legt eine virtuelle Treiber-CD ein. Nach der – automatisch gestarteten – Installation und einem Reboot funktioniert das virtuelle Netzwerk, der Mauszeiger ist nicht mehr gefangen, und das VirtualBox-Fenster lässt sich beliebig in der Größe verändern. Der besondere Clou ist jedoch, dass nach Drücken der rechten Steuerungstaste zusammen mit der Taste *L* VirtualBox in einen Seamless-Modus umschaltet: Die Fenster des XP-Gastes werden scheinbar nahtlos auf dem Linux-Desktop angezeigt. Allerdings ist dieser Modus nicht ganz perfekt: bei unserem Xubuntu-Testgerät erschien VirtualBox mit lediglich einem Icon in der Taskleiste und die Windows-Fenster blieben ständig im Vordergrund, eventuell geöffnete Linux-Fenster lagen dahinter. Einzelne Windows-Fenster lassen sich nicht auf andere virtuelle Desktops schieben. Wer für den Windows-Gast jedoch einen eigenen virtuellen Desktop einplant und auf Kleinigkeiten wie eine bei Nichtbenutzung versteckte Taskleiste achtet, profitiert vom Umschalten zwischen Wirt und Gast, das leichter als bei anderen Virtualisierungslösungen ist. Nahtlos ist auch die gemeinsame Nutzung von Ordnern des Hostsystems möglich. Eine Beispielfreigabe für den temporären Ordner entsteht im Verwaltungsfrontend der VirtualBox oder für die virtuelle Maschine mit





**Vorbildlich: Unter [www.virtualbox.org](http://www.virtualbox.org) finden Sie eine detaillierte Liste möglicher (und problematischer) Gastsysteme.**

Eine tolle Option für Administratoren ist, dass sich praktisch alle Funktionen der VirtualBox auf der Kommandozeile nutzen lassen. So können neue virtuelle Maschinen erzeugt, Shared Folders hinzugefügt oder einfach Maschinen gestartet werden. Gerade wer immer sein Windows XP offen haben möchte – seamless natürlich – kann in seine Xsession-Datei am Anfang den Startbefehl eintragen und beim Beenden automatisch mit

```
VBoxManage controlvm XP acpipowerbutton
```

die virtuelle Maschine herunterfahren. Daneben stehen eine Reihe von Subkommandos für die Konvertierung von Festplatten-Images oder zum Erstellen von Schnappschüssen zur Verfügung. Auch einige Netzwerkfunktionen, beispielsweise zum Aufbau eines Netzwerks zwischen allen virtuellen Maschinen eines Nutzers stehen nur über die Kommandozeile zur Verfügung.

Namen *XP* mit dem Befehl

```
VBoxManage sharedfolder add XP  
-name tmp -hostpath /tmp
```

Anschließend kann im Gastsystem im Explorer unter *Extras/Netzlaufwerk verbinden* dem unter `\\vboxsvl\tmp` verfügbaren Share ein Laufwerksbuchstabe zugewiesen werden.

nen sich Shares mounten lassen. Unter dem openSUSE 10.3 Gast ließen sich die Additions nach Installation von *kernel-source* und *gcc* mit dem mitgelieferten „Run-Script“ installieren. Leider schlugen beim openSUSE-Gast sowohl das Mounten des Shared Folders mit

```
mount -t vboxsf tmp /mountpoint
```

## Linux zu Gast

Häufig möchte man auch Linux als Gast nutzen – sei es um gefahrlos eine andere Distribution auszuprobieren oder um auch unter Windows mit typischen Linux-Applikationen zu arbeiten. Auch für Linux existieren Guest-Additions, mit denen nicht immer die rechte Steuerungstaste bemüht werden muss, um die Maus zu befreien und mit de-

als auch das automatische Befreien der Maus am Bildschirmrand fehlt. Da sich ältere Gäste problemlos konfigurieren ließen, vermuten wir einen Zusammenhang zwischen dem recht frischen SUSE-Kernel und den fast gleichzeitig mit openSUSE 10.3 erschienenen VirtualBox-Additions. Abhilfe sollte ein Vanilla-Kernel schaffen.

## Netzwerk-Feinheiten

Wie eingangs erwähnt, erfordert der Zugriff auf geöffnete Ports der virtuellen Maschinen die Einrichtung der Bridgetreiber. Dafür installieren Sie zunächst das Paket *bridge-utils*. Anschließend wird *eth0* mit einer Brücke verbunden. Die Brücke *br0* wiederum erbt die bislang für *eth0* getätigten Einstellungen. Die resultierende `/etc/network/interfaces`:

## So geht es: Installation von VirtualBox

➤ Am einfachsten ist die Installation von VirtualBox unter Debian und Ubuntu. Bei weiteren Debian-basierten Betriebssystemen ist die Verwendung der stark zugeschnittenen Installationsquellen mit Vorsicht zu genießen.

- Fügen Sie Ihrer `/etc/apt/sources.list` eine Repository-Zeile für den Innotek-Server hinzu. Für Ubuntu 7.10 (Gutsy Gibbon) erledigten Sie das mit dem Befehl:

```
sudo echo 'deb http://www.virtualbox.org/debian gutsy non-free' >>  
/etc/apt/sources.list
```

Je nach Version ist das Schlüsselwort *gutsy* zu ersetzen (*feisty* für Ubuntu 7.04, *dapper* für Ubuntu 6.06, *etch* für Debian 4.0 und *sarge* für Debian 3.1).

- Laden Sie den Signaturschlüssel von Innotek herunter und importieren Sie ihn:

```
cd /tmp/  
sudo wget http://www.virtualbox.org/  
debian/innotek.asc  
sudo apt-key add innotek.asc
```

- Aktualisieren Sie nun Ihre Installationsquellen und bringen Sie Ihr System auf den neuesten Stand:

```
sudo apt-get update  
sudo apt-get upgrade
```

Unter Umständen müssen Sie jetzt Ihren Rechner rebooten.

- Jetzt ist Ihr Ubuntu fit für VirtualBox. Überprüfen Sie zunächst die Verfügbarkeit:

```
sudo apt-cache search virtualbox
```

Es sollte Version 1.5.2 oder höher angezeigt werden.

Wird eine ältere oder die OSE-Version angezeigt, kann dies an einem Paketquellenkonflikt liegen: Möglicherweise haben Sie ein inoffizielles Backports-Repository aufgenommen. Stimmt die Version, starten Sie die Installation:

```
sudo apt-get install virtualbox
```

Um zu prüfen, ob beim Neustart alle Kernelmodule geladen werden, sollten Sie Ihren Rechner rebooten und dann ausprobieren, ob Sie VirtualBox starten können. Für die gängigsten Distributionen haben wir VirtualBox-Pakete im jeweiligen Paketformat beigelegt.

Sie können diese mit *dpkg -i paket.deb* respektive *rpm -Uvh paket.rpm* installieren. So kommen auch Anwender ohne schnelle Internetverbindung in den Genuss von VirtualBox.

```

auto lo
iface lo inet loopback
auto eth0
iface eth0 inet manual

auto br0
iface br0 inet dhcp
        bridge_ports eth0

```

Für das Hinzufügen virtueller Netzwerkkarten bringt *VirtualBox* eine Hilfsapplikation mit, der Sie den Namen des virtuellen Interfaces (meist *vbox0* bis *vbox1*) und den Benutzer übergeben, der auf dieses zugreifen darf (hier *mattias*):

```
VBoxAddIF vbox0 mattias br0
```

Jetzt können Sie in der Konfiguration Ihrer virtuellen Maschine den Typ der Netzwerkverbindung von *NAT* auf *Host* umstellen. Zudem müssen die Berechtigungen auf der Geräte-

datei */dev/net/tun0* so geändert werden, dass Nutzer, deren *VirtualBox* im Bridged-Modus auf das Netz zugreifen können, Schreibrechte haben.

Nach dem nächsten Start sind alle offenen Ports des Gastsystems von außen erreichbar: Dauerhaften Erfolg hatten wir bei manchen Gästen nur, wenn wir die IP-Adressen statisch vergaben und zunächst ausgehende Verbindungen (beispielsweise per *ping*) initiierten.

### Fazit

Von der Einfachheit der Konfiguration kommt *VirtualBox* nicht ganz an *VMware* heran, wenn es um die Netzwerk-Konfiguration geht, wer allerdings auf ein gebrücktes Netzwerk verzichten kann, hat mit *VirtualBox* unter Umständen schneller seine Gastsysteme eingerichtet als unter *VMware*. Gerade für den Heimnutzer ist die *VirtualBox* deshalb eine

sehr gute Alternative sowohl zum kostenlosen, aber im Funktionsumfang reduzierten *VMware-Player* als auch zu den vielen freien Virtualisierungslösungen, deren Flexibilität nicht an *VirtualBox* heranreicht. Dass *Innotek* mittelfristig auch die *VMware-Serverversionen* im Blick hat, zeigen die vielfältigen Kommandozeilenoptionen sowie die angekündigte Portierung auf einen eigenen Hypervisor/Microkernel.

In Zeiten steigender Energiepreise ist die Konsolidierung von Servern ein zukunfts-trächtiges Geschäft – allerdings muss bis dahin auch die noch die fehlende Unterstützung für Gäste mit PAE-Erweiterungen und 64-Bit implementiert werden.

Ein großer Wurf für Desktopanwender ist *Innotek* mit dem *Seamless-Modus* gelungen, der auf einfachste Art und Weise einzelne *Windows-Fenster* fast nahtlos auf dem *Linux-Desktop* integriert. jkn

## WEBLINKS

[www.virtualbox.org](http://www.virtualbox.org)

Auf der *Community-Seite* finden Sie *Downloads* in *Quellcode-* und *Binärform* sowie *konkrete technische Dokumentation* – und natürlich *Links zu Foren* und *Mailinglisten*.

[www.innotek.de](http://www.innotek.de)

Der *VirtualBox-„Erfinder“* und *Hauptentwickler Innotek* stellt auf seiner *Firmenseite* die *Konzepte von VirtualBox* und des *künftigen Hypervisor-Unterbaus* vor.

## Die Sache mit der Lizenz

Das Lizenzmodell, welches *Innotek* für *VirtualBox* gewählt hat, sorgt in *Internetforen* oft für *Verwirrung*. Tatsächlich wird *VirtualBox* unter einer *Doppellizenz* vertrieben: Zum einen sind die *Quellcodes* der fast kompletten *Virtualisierungssoftware* unter der *GNU General Public License* erhältlich, zum anderen bietet *Innotek* die *kommerzielle Lizenzierung* der *Software* an. Weil *Innotek* alle Rechte am *Quellcode* hält, bestimmt der *Softwarehersteller* die *Verwertungsformen* und kann auch *scheinbar widersprüchliche Lizenzmodelle* parallel anbieten.

Alle *Binaries* von der *Innotek-Webseite* und unserer *Heft-DVD* unterliegen daher *Innoteks „VirtualBox Personal Use and Evaluation License“*. Diese besagt, dass *Privatpersonen VirtualBox* ohne *Einschränkungen* nutzen dürfen, *Unternehmen* und *Behörden* die *Software* aber nur zu *Testzwecken* einsetzen dürfen. Einen *Testzeitraum* nennt *Innotek* nicht. Preise für den *kommerziellen Einsatz* teilt *Innotek* auf *Anfrage* mit. Sie können diese *Einschränkungen* ganz *legal* umgehen, indem Sie die *Quellcodes* der *Open Source Version* selbst *kompilieren*. *Innotek* stellt keine *unüberwindlichen Hürden* hierfür auf, sondern zeigt die *Vorgehensweise* und *benötigte Systemvoraussetzungen* sehr gut auf der *Webseite [www.virtualbox.org](http://www.virtualbox.org)*. Dem *Ergebnis* fehlen lediglich *wenige Funktionen*, die vor allem für den *Betrieb in Rechenzentren* sinnvoll sind. So erzeugte *Binärdateien* dürfen Sie *unternehmensweit* einsetzen oder auch *weitergeben*. Bei der *Weitergabe* sind dann gemäß der *GPL Quellcodes* mitzuliefern, aber auch die *Markenrechte* an *VirtualBox* zu beachten. Aufgrund der *Verfügbarkeit* einer *Open-Source-Edition* gehen wir davon aus, dass *künftig Distributoren* verstärkt dazu übergehen, *VirtualBox* im *Standardumfang* der *Distribution* mitzuliefern.

## VirtualBox unter Windows

Auch mit *Windows* als *Wirtssystem* lässt sich *VirtualBox* einsetzen. Sie können damit beispielsweise die *Distributionen* der *Heft-DVD* testen, ohne Ihre *Festplatte* neu *partitionieren* zu müssen, denn schließlich wir nur mit einer *Image-Datei* auf einer *NTFS-Partition* gearbeitet. Damit Sie den *Bridged-Modus* verwenden können, mit dem die *virtualisierten Linux-Instanzen* direkt in Ihrem *Ethernet-Netzwerk* *kommunizieren* können, ist jedoch ein *Ausflug* auf die *Kommandozeile* nötig.

- Installieren Sie *VirtualBox* per *Doppelklick* aus dem *MSI-Paket* von der *Heft-DVD*. Während der *Einrichtung* müssen Sie die *Installation* zweier nicht *signierter Treiber* bestätigen.

- Öffnen Sie die *Windows-Eingabeaufforderung (Start/Ausführen/command.com)* und wechseln Sie in dieser in das *VirtualBox-Installationsverzeichnis*:

```
cd "C:\Programme\innotek VirtualBox"
```

- Starten Sie hier die *Installation des Bridge-Treibers*:

```
VBoxManage.exe createhostif "VM1"
    external"
```

Nach einigen *Sekunden* müssen Sie die *Installation eines nicht signierten Treibers* bestätigen. Wenn Sie jetzt *virtuelle Maschinen* erstellen oder *abändern*, können Sie als *Netzwerkschnittstellentyp* *Hostinterface* und als *Name* *VM1 external* auswählen.