Ubuntu Linux in einer virtuellen Box installieren

Wer gerne Linux testen möchte, sich aber vor einer „Echtinstallation“ scheut, für den ist diese Anleitung gedacht. Es besteht keine Gefahr für das bereits vorinstallierte Windows, weil an der Partitionierung der Festplatte nichts geändert wird. Trotzdem empfiehlt sich vorher eine Daten- und Systemsicherung, was man aber sowieso regelmäßig machen sollte.

Um Linux und dessen Installation kennenzulernen, kann man es anhand einer virtuellen Maschine versuchen. Dabei wird ein PC im PC gestartet, der sich etwas von dem RAM nimmt und eine Datei dient als Festplatte. Man kann auch – genügend Arbeitsspeicher auf dem „echten“ PC (= Host) vorausgesetzt – zwei oder sogar mehr solcher Maschinen (= Guests) gleichzeitig starten und Daten austauschen (z.B. Ubuntu 32Bit, Ubuntu 64Bit und zusätzlich noch ganz die neueste Version zum Testen).

Das Programm „Virtual Box“ von Oracle downloaden und installieren. Sicherstellen, dass im Bios die Hardware-Virtualisierung aktiviert worden ist (ohne Hardware-Virtualisierung sind nur 32Bit-Gäste möglich und diese laufen dann langsamer). Einige ältere PCs und auch solche mit einem Intel-Atom-Prozessor bieten diese Hardware-Virtualisierung aber nicht an, obwohl das System durchaus mit 64Bit-Betriebssystemen arbeiten kann. Auf solchen Systemen sind keine 64Bit-Gäste möglich.

Nach der Installation des Programmes „Neu“ wählen, um die virtuelle Maschine anzulegen. Linux Betriebssystem und Ubuntu (bei 32Bit Ubuntu) bzw. Ubuntu 64Bit wählen.

Die Vorgaben mit 512 MB RAM und 8 GB Festplatte ev. erhöhen auf 768 bis 1024 MB RAM. Zumindest während der Installation, um diese zu beschleunigen, wenn das System über ausreichend RAM (ab 2 GB) verfügt. Für die Festplatte gleich 16 bis 24 GB reservieren, denn diese wächst dynamisch und belegt nicht den ganzen Platz auf einmal. Ein nachträgliches Vergrößern ist relativ schwierig.

Es empfiehlt sich die Festplatten-Container-Datei in einem leicht zugänglichen Verzeichnis ablegen und nicht in den Untiefen der "Dokumente und Einstellungen".

Virtuelle Maschine starten, von CD/DVD booten wählen und die Ubuntu-ISO-Datei mounten. Jetzt sollte die CD/DVD booten. Installieren wählen, Sprache auswählen, Zeitzone … usw.

Die rechte STRG-Taste befreit die gefangene Maus, falls man zwischendurch mit einem anderen Programm arbeiten möchte.

Bei der Frage nach der Festplattenaufteilung diese am Besten dem System überlassen.

Bei den Benutzerangaben:

Passwort gut merken, das brauchen wir später -> [Passwort eingeben] bezieht sich darauf. Benutzer automatisch anmelden ist eine gute Option; das /home-Verzeichnis braucht man aber nicht zu verschlüsseln.

Die Installation dauert so ca. 30 – 60 Minuten. Leider ist besonders abends die Downloadrate der Updates ziemlich langsam.

Virtual Box Additions installieren

Wenn das System gestartet ist, dann links oben auswählen

Anwendungen -> Zubehör -> Terminal
(mit rechter Maustaste „als Starter der Arbeitsoberfläche hinzufügen“ ist praktisch)

In dieser Eingabeaufforderung

sudo –i eingeben dann
[Passwort] eingeben

Jetzt hat man für 15 Minuten Rootrechte = Administrator.

Die rechte STRG-Taste befreit die gefangene Maus; Menü von VirtualBox „Geräte“ wählen und dann Gasterweiterungen installieren.

cd / (in Linux / entspricht in Windows \)

(ev. muss das Verzeichnis cdrom noch mit mkdir cdrom angelegt werden)

mount /dev/cdrom /cdrom (Meldung kommt, das im readonly Modus gemountet wurde)
cd /cdrom

dir bzw.
ls -l zeigt das Verzeichnis
ls -lai zeigt auch versteckte Verzeichnis (die beginnen mit einem .)

./VBoxLinuxAdditions-x86.run (bzw. ./VBoxLinuxAdditions-amd64.run bei 64Bit)

Sollt man nur ein leeres Verzeichnis sehen, dann hilft es, die virtuelle Maschine nochmals zu booten und dann ins Verzeichnis /media/VBOXADDITIONS..(Versionscode) zu wechseln

cd /media
cd VBOX -> Tab-Taste drücken und Name wird automatisch ergänzt

dann kann schon ./VBoxLinuxAdditions-xxx.run gestartet werden

Info für Kubuntu:
zuerst gcc (apt-get install gcc) installieren, dann erst die VBoxLinuxAdditions.
Sonst können diese nicht compiliert werden und funktionieren dann nicht.

Die Installation dauert etwas, dann mit

depmod-aeq aufräumen

Virtuelles Linuxbetriebssystem niederfahren z.B. mit

shutdown -h now auf der Console eingeben.

shutdown -h now = Betriebssystem niederfahren und virtuellen Rechner ausschalten
shutdown -r now = Betriebssystem niederfahren mit anschließendem Neustart
(hier genügt einfach reboot auf der Console zu schreiben)

Info:

Nach einem (Online-) Update des Linuxsystems (z.B. über die Paketverwaltung) oder nach einem Update auf eine neuere Version von Virtual Box kann es notwendig sein, dass man die Additions nochmals installieren muss. Man bemerkt es, wenn der Mauszeiger eingefangen wird. Das Script deinstalliert zuerst die vorhandenen Gasterweiterungen und installiert dann neue.

Wenn das virtuelle Linux niedergefahren ist, dann bei der Virtualbox auf Gemeinsame Ordner klicken (ganz unten bei den Einstellungen) und auf das + Symbol rechts oben klicken und einen Ordner der Festplatte hinzufügen.

Dieser dient zum Datenaustausch mit der virtuellen Maschine. Ordner-Name merken! [Ordner] bezieht sich darauf.

Die virtuelle Maschine erneut booten, dann wieder das Terminal aufrufen und sich root-Rechte verschaffen. Jetzt funktioniert auch copy & paste zwischen Host und Gast.

Installation von Softwarepaketen wie emacs, wine, dosbox, dosemu usw.

Einfach zuerst mal z.B. wine eingeben. Dann sagt Ubuntu, wie man sich dieses Paket holen kann; nämlich mit

apt-get install [Paketname] (so erfährt man auch gleich den aktuellen Paketnamen)

Man kann nur ein Paket zur gleichen Zeit downloaden und installieren.
Tipp: Nicht im Hintergrund mit & ausführen, da apt-get tw. Usereingaben benötigt
(sonst hängt das Programm)

Netzlaufwerk zum Datenaustausch mit dem Hostcomputer mounten

cd /
mkdir netz (irgendein Name für den Ordner der dann das Netzlaufwerk aufnehmen soll)
cd netz
mount -t vboxsf [Freigabename für Ordner auf Host] /netz

Netzlaufwerk automatisch beim Neustart mounten:
(damit man das nicht nach jedem Start manuell erledigen muss)

Die Datei /etc/init.d/rc.local mit dem Editor emacs öffnen

zur Not mit wine und einem Windows-Editor z.B. Notepad

wine notepad rc.local

(Menü Bearbeiten -> Schriftarten einstellen auf Courier New ist praktisch)
(funktioniert in jedem Verzeichnis, da wine das notepad „bereits eingebaut“ hat)

Nach Path einfügen (in meiner Installation steht)

PATH=/sbin:/usr/sbin:/bin:/usr/bin

mount -t vboxsf [Freigabename für Ordner auf Host] /netz

Windowsprogramme mit wine ausführen

wine Dateiname[.exe] [opt. Programmparameter Windowsprogramm] &

z.B. wine notepad textdat.txt

startet das Windowsprogramm; & führt es im Hintergrund aus
-> Konsole bleibt frei für weitere Kommandos.

Konfiguration von wine mit

winecfg &

DOS-Programme mit dosbox ausführen

dosbox &

Alle weiteren Kommandos beziehen sich jetzt auf die Dosbox:

mount c /netz (englisches Tastaturlayout / entspricht -)

c: (englisches Tastaturlayout : entspricht ö)

So jetzt hat man einen 386er / 486er mit MS-DOS vor sich. Den deutschen Tastaturtreiber noch aufrufen, der sich (hoffentlich) auf dem Laufwerk befindet.

Jetzt in der Dosbox eingeben:

config -writeconf .dosbox/dosbox.conf

(englische Tastaturlayout, x und y sind vertauscht sowie - = / sowie ß = -)

(ev. kommt hier bei neueren Versionen eine Fehlermeldung; kein Problem,
einfach weitermachen, weil dann existiert schon eine Konfigurationsdatei)

dosbox beenden (exit eingeben)
in der Console eingeben:

cd
cd .dosbox
dir (zeigt ob die Datei dosbox.conf oder z.B. dosbox-0.73.conf heißt)
emacs dosbox.conf & (ev. emacs dosbox-0.73.conf & je nach Version)

oder halt alternativ

wine notepad dosbox.conf &

ziemlich unten ändern auf

keyboardlayout=gr (damit die Tastur gewohntes deutsches Layout bekommt)

ganz unten Abschnitt [autoexec]

mount c /netz (Ordner /netz wird Laufwerk C:)
c:

im Menü file – save zum Speichern und dann file – quit zum Beenden

dosbox neu von der Console aus starten und schon sollte alles laufen

Das muss für jeden User separat geschehen, z.B. auch für den root-User
und den „normalen“ User falls beide deutsche Tastatur + Laufwerk C: möchten
Weitere Infos unter http://wiki.ubuntuusers.de/Spiele/DOSBox

DOS-Programme mit dosemu ausführen

mit sudo -i root-Rechte sichern (dosemu funtioniert nicht unter eingeschränkten Userrechten)

dosemu & oder
xdosemu & zum Starten das Programmes ("gui-Version")

die Tastatur sollte (hoffentlich) deutsch sein (sonst Konfigurationsdatei bearbeiten)

Das Laufwerk E: entspricht standardmäßig dem Linux Rootverzeichnis /

exitemu beendet das Programm (Info: exit bewirkt nichts)

Die Konfigurationsdatei kann wie folgt bearbeitet werden:

cd /etc/dosemu

emacs dosemu.conf

(oder alternativ: wine notepad dosemu.conf)

Weitere Infos unter http://wiki.ubuntuusers.de/Dosemu

Adobe Flashplayer Plugin (benötigt für Youtube)

Adobe Flash kann von folgender Seite heruntergeladen werden:

http://get.adobe.com/de/flashplayer

es gibt sogar eine Version für 64Bit Linux:

http://labs.adobe.com/downloads/flashplayer10_square.html

Die mit ".tgz" gepackte Linuxversion mit dem Typ ".so" auswählen

Download - Firefox speichert die Downloads standardmäßig im Ordner
/home/<username>/Downloads z.B.
/home/franz/Downloads

ins Verzeichnis /usr/lib/mozilla/plugins wechseln mit

cd /usr/lib/mozilla/plugins

cp /home/<username>/Downloads/<filename>.tgz .

entpacken mit

tar xvzf <filename>.tgz

das tgz-File kann dann mit

rm <filename>.tgz

gelöscht werden.

Firefox neu starten und schon sollte es funktionieren

„Nette“ Plugins für Firefox:

https://addons.mozilla.org/de/firefox/addon/forecastfox-weather/

https://addons.mozilla.org/en-US/firefox/addon/flashblock/

Eigene Linuxanwendungen mit GTK+ compilieren

Die Installation von GTK+ erfolgt mit

apt-get install libgtk2.0-0 libgtk2.0-dev

Compilieren von GTK+ Beispielen mit

gcc -o gtkafritz gtkafritz.c `pkg-config --libs --cflags gtk+-2.0`

(ev. noch der Parameter -Wall für noch höhere Warnstufe)

Ausführen des compilierten Programmes mit

./gtkafritz &

Auf einem 32Bit-System wird ein 32Bit-Linux-Programm erstellt; dieses läuft auch auf 64Bit-Systemen. Ein 64Bit-System erstellt ein 64Bit-Linux-Programm; dieses läuft nicht auf 32Bit-Systemen. Auf anderen oder älteren Distributionen kann es laufen, aber eine ältere Version der glibc auf diesen Systemen kann dies auch verhindern. (Fehlermeldung kommt, dass glibc in der Version x.x benötigt wird.)

Mann kann auch mit einem 64Bit-System eine 32Bit-Anwendung compilieren, wenn man das 32Bit-Compiler-Package installiert (siehe weiter unten im Abschnitt wine) – kurz erklärt mit

apt-get build-dep wine1.2 (etwas zeit- und platzintensiv)

downloaden und dann compilieren mit

gcc -m32 … <weitere Optionen>

Festplattenplatz einsparen

Die downgeloadeten Updates benötigen auch Platz, der teilweise leicht wieder freigemacht werden kann. Die gespeicherten .deb – Pakete können mit folgendem Befehl gelöscht werden:

apt-get clean (root Rechte nötig, sonst mit sudo apt-get clean – Root-Passwort Eingabe nötig)

Das kann einige 100 MB an Platz bringen.

Weiters können noch ältere parallel installierte Linux-Kernel gelöscht werden.
Da dies aber etwas riskant ist (man kann auch irrtümlich den aktuellen Kernel löschen), empfiehlt sich eine Sicherheitskopie der Containerdatei vor dieser Aktion.

Auf der Arbeitsoberfläche links oben
System -> Systemverwaltung (Administration bei engl. Version)
-> Synapitcs Paketverwaltung

auswählen und nach Linux Kernel suchen. Dann nach der ersten Spalte sortieren und an der Versionsnummer orientieren. So kann man ältere Linux-Kernel deinstallieren.

Videoclips anschauen (z.B. DivX-Videos)

einfach Doppelklick im Nautilis ("Explorer") der AVI-Datei
meldet, dass noch ein Codec installiert werden muss
Installation bestätigen, Root-Passwort eingeben,
Download der Pakete erfolgt und schon sieht man das Video

Netzwerk ist in der virtuellen Maschine nicht verfügbar

Es kann nach dem Starten der virtuellen Maschine durchaus vorkommen, dass das Netzwerk nicht verfügbar ist. Somit kann man nicht auf das Internet zugreifen.

Dieses Problem ist aber relativ einfach zu lösen:

In der rechten oberen Ecke sieht man ein "Wellensymbol" (mehrere Halbkreise) mit einem roten Rufzeichen. Auf dieses 1 x mit der rechten Maustaste drücken und dann das Häkchen vor "Netzwerk aktivieren" durch Klick mit der linken Maustaste entfernen. Gleich danach nochmals mit der rechten Maustaste darauf klicken und wieder "Netzwerk aktivieren" mit der linken Maustaste anhaken.

Dann sollte kurz danach eine symbolische Netzwerkbuchse erscheinen und die Netzwerkverbindung wieder funktionieren.

wine und wine64 selber compilieren

Achtung: vorher Datensicherung durchführen (Containerdatei sichern)!

Mit wine64 können 64Bit-Windows-Anwendungen ausgeführt werden; allerdings nur mit der 64Bit-Ubuntu-Version, nicht mit der 32Bit-Version (sollte aber irgendwie einleuchtend sein).

Alle Befehle müssen als Administrator (sudo -i) ausgeführt werden

Download der notwendigen Pakete inkl. dem 32Bit Compilerpaket (unter 64Bit kann man sonst nur standardmäßig 64Bit-Software compilieren, nicht aber 32Bit-Software)

apt-get build-dep wine1.2 (dauert etwas)
Parallel dazu kann schon das wine-Sourcepaket heruntergeladen werden:

http://sourceforge.net/projects/wine/files/Source/

man muss etwas nach unten scrollen um die aktuelle Version zu finden -
zum Zeitpunkt der Erstellung dieser Anleitung war folgende Version aktuell:

http://sourceforge.net/projects/wine/files/Source/wine-1.3.28.tar.bz2/download

(also die Version 1.3.28)

Sofern vorhanden muss man noch die mit apt-get installierte wine-Version entfernen:

apt-get remove wine1.2

Sicherheitscheck mit

which wine

Es darf keine Datei "wine" gefunden werden; falls doch, dann diese löschen.

Weiters müssen auch noch die Konfigurationsdateien im HOME-Verzeichnis entfernt werden:
mit cd ins HOME-Verzeichnis wechseln, dann mit

cd .wine ins .wine Verzeichnis wechseln

alle Dateien und Unterverzeichnisse löschen mit

(Vorsicht mächtiger Befehl, sicherstellen dass man im richtigen Verzeichnis ist z.B. mit pwd)

rm -R * .* das Leerzeichen zwischen * .* nicht vergessen

Das leere .wine Verzeichnis noch entfernen mit

cd wieder ins HOME-Verzeichnis und
rmdir .wine

mit ls -lai noch nachschauen, ob ev. noch andere Verzeichnisse vorhanden sind, die auch mit .win oder .wine beginnen. Auch diese nach dem oben genannten Verfahren entfernen.
Jetzt kann man schon mit dem Compilieren beginnen:

cd / Wechsel ins Root-Verzeichnis

mkdir wine das Arbeitsverzeichnis erstellen (hier /wine)

in dieses dann den Sourcecode downloaden oder hineinkopieren

bunzip2 wine-1.3.28.tar.bz2 entpacken von bz2 zu tar

tar vxf wine-1.3.28.tar entpacken des tarballs

(nach dem erfolgreichen Auspacken kann mit rm wine-1.3.28.tar das tar-Archiv gelöscht werden; weiters nach dem erfolgreichen Compilieren und Installieren das ganze Verzeichnis)

cd wine-1.3.28 das ist jetzt das neue Arbeitsverzeichnis

Vorbereitung zur Compilierung:

make distclean (bei frisch entpackter Source nicht notwendig)
unset CFLAGS
unset LDFLAGS
export CFLAGS="-m32"
export LDFLAGS="-L./lib32"

so jetzt kann compiliert werden; jeder Schritt benötigt aber mehrere Minuten:

./configure
make (das eigentliche Compilieren ist am zeitintensivsten)
make install

nach der Compilierung der 32Bit-Version wird jetzt die 64Bit-Version erstellt:

make distclean (jetzt aber unbedingt notwendig)
unset CFLAGS
unset LDFLAGS

./configure --enable-win64 (zwei Bindestriche vor enable)
make (wieder am zeitintensivsten)
make install

Test, ob wine und wine64 im Pfad verfügbar sind:

which wine64

jetzt sollte man sehen:

/usr/local/bin/wine64

(das gleiche mit wine wiederholen)

Zuerst eine (am besten kleine und unkomplizierte) 32Bit-Windowsanwendung starten mit

wine <Name der Anwendung> <optionale Parameter für die Windowsanwendung>

Beim ersten Start wird die Arbeitsumgebung wieder aufgebaut Es kommt eine Meldung, dass der Gecko-Font noch heruntergeladen werden muss; diese noch bestätigen.

Kurze Zeit später sollte dann (hoffentlich) die Windows-Anwendung starten.

So jetzt kann der erste Versuch gemacht werden, eine 64Bit Anwendung zu starten:

wine <Name der Anwendung> <optionale Parameter für die Windowsanwendung>

(64Bit-Windows-Anwend. gibt es z.B. unter http://members.aon.at/fraba/download.htm)

Mit wine sollte man jetzt sowohl 16Bit/32Bit- als auch 64Bit-Windowsanwendungen starten können – das kann direkt kein „richtiges“ Windows.

Folgende Anleitung ist nur für den Fall, dass keine 64Bit-Windowsanwendungen gestartet werden können:

Falls Probleme mit 64Bit-Anwendungen auftreten, zuerst mit wine64 anstatt wine versuchen.

Sollte das nichts (mehr) helfen, dann bleibt noch folgender Versuch:

Wechsel in das Verzeichnis wo wine64 drinnen ist:

cd /usr/local/bin

neues "Batch-File" erstellen mit

emacs wine_64

(nicht mit wine notepad wine_64 weil sonst wird am Zeilenende ein CTRL-M angehängt und das Script kann das Programm wine64 nicht starten)

Exakt diesen Text (4 Zeilen inkl. einer Leerzeile) eingeben und speichern:

```
#!/bin/bash
#Programm Wine64
```

WINEPREFIX=~/.wine64 wine64 $1 $2 $3 $4 $5 $6 $7 $8 $9

(fürs erste tuts auch bis $4 oder $5 für die Argumente)

Das Script ausführbar machen mit

chmod a+x wine_64

64Bit-Windowsanwendungen aufrufen mit

wine_64 <Name der Anwendung> <optionale Parameter für die Windowsanwendung>

16Bit/32Bit-Windowsanwendungen aufrufen mit

wine <Name der Anwendung> <optionale Parameter für die Windowsanwendung>

Weiterführende Links zu wine und Problemlösungen:

http://wiki.winehq.org/Wine64

http://wiki.winehq.org/WineOn64bit

http://wiki.winehq.org/Recommended_Packages

https://wiki.archlinux.org/index.php/Wine#Using_WINEARCH

http://www.cyberciti.biz/tips/compile-32bit-application-using-gcc-64-bit-linux.html

http://forum.winehq.org/viewtopic.php?p=47329&sid=f77260a174683e550ac95775134e6441

Rechnernamen nachträglich ändern

So jetzt hat man das komplett perfekte System fertig und kommt darauf, dass der Name des Rechners nicht mehr als passend empfunden wird. Der Rechnername kann ganz einfach durch Editieren der Datei /etc/hostname angepasst werden; danach neu booten.

emacs /etc/hostname (natürlich wieder mit Root-Rechten)

Info zur Festplattenplatzersparnis auf dem Hostsystem

Die Container Datei mit dem virtuellen Linux belegt nach der Installation ca. 3 - 4 GB.
Der Platzbedarf auf der Platte kann durch das Aktivieren des Attributes (NTFS) Komprimierung auf ca. die Hälfte reduziert werden. Dies sollte aus Performancegründen aber erst nach der Installation geschehen.

ERROR: X: Can't open display ":0.0"

Falls diese Fehlermeldung erscheinen sollte, wenn z.B. wine unter Kubuntu starten möchte, dann hilft folgende Vorgangsweise:

CTRL + ALT + F2 gleichzeitig drücken

damit kommt man zu einer Console mit einem login:

normal mit dem Benutzernamen und Passwort einloggen und
sudo –i für Administratorrechte

eine neue X-Session mit

xinit -- :1

und dann eine neue KDE-Sitzung starten mit

startkde

Es erscheint noch ein Fenster mit der Frage, ob dieser Server auch standardmäßig verwendet werden soll. Das noch bestätigen

QEMU

ist auch ein Emulator für einen PC und dieser läuft sogar innerhalb einer virtuellen Maschine, wenn auch ziemlich langsam. Die Installation erfolgt mit

apt-get install qemu

Beispiel für eine Befehlszeile zum Starten einer virtuellen Maschine

qemu -L . -m 512 -hda windowsxp.vmdk -localtime -cdrom winxp.iso -net nic,model=ne2k_pci -net user -usb -usbdevice tablet [-boot d]

-m 512	512 MB Arbeitsspeicher für die virtuelle Maschine zur Verfügung stellen
-hda windowsxp.vmdk	Name des Festplattenimages
-cdrom winxp.iso	Name des CD/DVD-Images (bei Bedarf)
-localtime	Virtuelle Maschine läuft mit der lokalen Zeit
-net nic,model=ne2k_pci -net user	Netzwerkoptionen, hier Netzwerkkarte NE 2000 PCI
-usb -usbdevice tablet	so wird der Mauscursor nicht eingefangen
-boot d	vom D-Laufwerk = hier CD/DVD-Image booten (benötigt für die Installation eines Betriebssystems von CD)
-win2k-hack	nur für die Installation von Windows 2000 (während der Installation; nachher nicht mehr)

Ein eingefangener Mauscursor kann durch gleichzeitiges Drücken von STRG-ALT wieder befreit werden (mit -usb -usbdevice tablet sollte dies nicht nötig sein)

Ein leeres Festplattenimage wird folgendermaßen erstellt:

qemu-img create -f qcow2 <namederimagedatei>.vmdk 4000M

Hier wird ein 4000 MB großes Festplattenimage erstellt, welches aber nicht sofort den ganzen Speicherplatz benötigt, sondern dynamisch bei Bedarf wächst.

QEMU selber compilieren

Auch das geht relativ einfach; die Source herunterladen und z.B. ins Verzeichnis

/qemu (cd / dann mkdir qemu und dann cd qemu) entpacken mit

tar xvzf qemu-0.14.1.tar.gz (dzt. aktuelle Version)

cd qemu-0.14.1

Zum Compiliern notwendige Pakete downloaden mit:

apt-get install libsdl1.2-dev libsdl1.2debian

compilieren mit

./configure
make (das ist wieder am zeitintensivsten)
make install

Testen, ob qemu im Pfad verfügbar ist mit

which qemu

Links zu qemu:

http://www.computerleben.net/artikel/Windows_emulieren-238.html

http://wiki.ubuntuusers.de/qemu

http://qemu-buch.de/de/index.php/QEMU-KVM-Buch/_Speichermedien/_Images_anlegen

http://chris.silmor.de/slides/qemu.pdf

http://www.linuxforen.de/forums/showthread.php?t=141201

http://www.h7.dion.ne.jp/~qemu-win/Audio-en.html