BIC

WINE

Ein Windows Emulator für Unix

von Marcel Stanitzki

1 Was ist WINE ?

WINE ist eine freie Implementation der Windows API[1] für Unix-Plattformen. WINE besteht aus 2 Teilen, der eigentlichen Emulator *wine* und der *libwine*, mit der es möglich ist, Windows Software unter jeder Unix-Plattform zu compilieren und zu linken. Der Emulator ist nur für x86-basierende Unixe wie Linux 2.x.x, Solaris/X86 ab 2.5 und FreeBSD/OpenBSD/NetBSD verfügbar. Der Emulator läuft nicht auf nicht-X86-Plattformen wie UltraSPARC, Alpha oder PowerPC. Unterstützung für solche Plattformen ist erst in Planung. WINE unterstützt momentan die Emulation der Win16 API (Windows 3.0/3.1/3.11) und der Win32 API (Windows 95/98 Windows NT 3.51/4.0).

2 Welche Programme laufen mit wine ?

WINE befindet sich momentan noch im ALPHA-Stadium und wird als developer's release bezeichnet. Das bedeutet, man kann die Software testen und ausprobieren, man sollte sich aber über Misserfolge nicht wundern. Aktuelle Erfolge kann man im Internet (siehe Kapitel 6) nachlesen. Aus eigener Erfahrung und mehreren wine-Testsessions lässt sich folgendes sagen: Ältere Programme, die auf der Win16-API aufsetzen, laufen meist mehr oder weniger problemlos. Bei der Win32 API laufen kleinere Programme wie Solitaire oder Tools wie Graphic Workshop fast problemlos. Mit grösseren Paketen wie dem aktuellen Office oder der aktuellen Smartsuite sollte man sich keine so grossen Hoffnungen machen. Diese Programme laufen ziemlich eingeschränkt und stürzen meist nach kurzer Zeit ab. Wer allerdings eine ältere Version von Office oder Smartsuite sein eigen nennt, der kann mit wesentlich mehr Erfolg rechnen. Kurz, etwas ältere Win32-Programme sind mit etwas Geduld zum stabilen Arbeiten zu bewegen. Bei Spielen ist die Situation momentan so,dass alle Spiele, die unter NT laufen, auch mit großer Wahrscheinlichkeit unter WINE laufen (Starcraft ist so ein Fall). Der Multimedia und DirectX-Support für die Windows-spezifischen Multimedia-Funktionen ist im Entstehen. Zusammenfassend lässt sich sagen :

- mit Win16 Programmen gibt es fast keine Probleme mehr
- Programme die OLE[2] oder OLE 2 verwenden, sind problematisch
- Die meisten DirectX-Spiele laufen nicht, ebenso viele Multimedia-Tools,allerdings gibt es hier von Version zu Version Verbesserungen
- Programme, die direkt auf die Hardware zugreifen, oder herstellerspezifische Windows-VXD's[3] benutzen, werden nicht unterstützt. Ein Beispiel dafür wären Videokarten-Software oder Tools, die spezielle Schnittstellenkarten benötigen.
- Die Druckerunterstützung ist nur rudimentär vorhanden

[1] Die Schnittstelle zwischen Windows-Programmen und dem eigentlich Windows-System

[2] Object Linking and Embedding, ein Mechanismus von Windows, der das Einfügen von Daten aus anderne Windows-Programmen ins aktuelle Programm ermöglicht. Details siehe AnhangA

[3] Windows-Gerätetreiber werden als VXD bezeichnet

3 Download, Compilieren und Installieren von WINE

WINE wird zur Zeit offiziell nur als Source-Paket ins Netz gestellt. Binaries als rpm-Paket werden meistens von WINE-Usern zusammengestellt und werden nicht von den Developern unterstützt. Die Benennung der WINE Source-Pakete hat folgendes System : *Wine-YYMMDD.tar.gz*, wobei *YY* für das Jahr, *MM* für den Monat und *DD* für den Tag. Wine-990225.tar.gz ist also das Release vom 25.02.99. Die Source-Pakete werden zuerst auf diesen 4 FTP-Servern abgelegt.

```
metalab.unc.edu/pub/Linux/ALPHA/wine/development/Wine-990225.tar.gz
tsx-11.mit.edu/pub/linux/ALPHA/Wine/development/Wine-990225.tar.gz
ftp.infomagic.com/pub/mirrors/linux/sunsite/ALPHA/wine/
development/Wine-990225.tar.gz
ftp.progsoc.uts.edu.au/pub/Wine/development/Wine-990225.tar.gz
```

Alle FTP-Server, die tsx11 und sunsite spiegeln (Mirror!) haben die neuesten WINE-Pakete meist nach 1 bis 2 Tagen nach dem Release. tsx11 und sunsite gehören übrigens zu den am häufigsten gespiegelten Servern im Internet. Auch der FTP-Server der Uni Karlsruhe (ftp.rz.uni-karlsruhe.de) spiegelt sowohl sunsite als auch tsx11. Üblicherweise liegen die gespiegelten Server-Inhalte im Verzeichnis */mirrors* oder */pub/mirrors*. Das Source-Paket hat zur Zeit eine Grösse von ca. 2.8 MB. Zum Compilieren der Sourcen unter Linux braucht man

- Den aktuellen gcc oder egcs, also 2.7.2.3 oder 2.91.60
- Die libXpm (befindet sich bei S.u.S.E im Paket xpm in der Serie x)
- Die Tools flex und yacc/bison (diese Tools befinden sich bei der S.u.S.E in den Paketen flex und bison in der Serie d
- ca. 60-100 MB freien Platz auf der Festplatte.
- 48-64 MB Arbeitsspeicher sind empfehlenswert, um den Emulator auszuführen.

Zum Entpacken der Sourcen benutzt man folgenden Befehl

tar -xzf Wine-YYMMDD.tar.gz

Nach dem Entpacken befindet sich im aktuellen Verzeichnis ein Unterverzeichnis wine-YYMMDD, das die Sourcen enthält. Man wechselt in diese Verzeichnis. Dort befindet sich im Unterverzeichnis *documentation* jede Menge Lesenwertes über Wine. Ausserdem enthalten die Dateien *ANNOUNCE* und *README* Informationen zum aktuellen Release und eine kurze Anleitung zu WINE. Jetzt müssen die Sourcen nur noch konfiguriert werden.Dabei werden die Pfade zu den Includes und Libraries auf das aktuelle System angepasst und der Compiler und Linker werden auf ihre Funktionalität überprüft.Mit

./configure

werden diese Einstellungen automatisch vorgenommen. Wenn die Konfiguration erfolgreich war (Meldung : configuaration finished),müssen noch die Dependencies in den Makefiles[4] erstellt werden. Dies geschieht mit

[4] Ein Makefile enthlt die Anweisungen, wie das Programm zu compilieren und zu linken ist

make depend

Danach kann WINE compiliert werden, dies dauert je nach Rechner zwischen 10 und 30 Minuten. Die Compilation wird mit

make

gestartet. Wenn alles glatt gegangen ist, kommt die Meldung "wine build complete" . Jetzt können die libwine.a,wine, die manpages und die Konfigurationsdatei wine.conf (siehe 4) Dies geschieht per default in das Verzeichnis */usr/local* und die darunterliegenden Verzeichnisse *bin*,*lib*, *man* und *etc*. Um die Dateien zu installieren,braucht man root-Rechte. Die Installation wird mit

make install

gestartet. Wenn man WINE nochmals compilieren will, oder aber die Grösse des Source-Verzeichnisses reduzieren will, kann man mit

make clean

alle Object-Files löschen. Jetzt sollte noch überprüft werden, das */usr/local/bin* in der PATH-Variable enthalten ist. (Dies ist bei S.u.S.E der Fall). Damit ist die Installation beendet

4 Die /usr/local/etc/wine.conf

Jetzt muß der Emulator wine noch konfiguriert werden. Dies geschieht mit Hilfe der Datei *wine.conf*. Mit make install wird bei der Erstinstallation eine default wine.conf installiert, die nur noch angepasst werden muss.Die Datei hat eine INI-ähnliches Syntax. In der ersten Sektion werden die Laufwerksbuchstaben zugewiesen.

```
[Drive C]
Path=/dos/c
Type=hd
Label=MS-DOS
Filesystem=win95

[Drive I]
Path=/cdrom
Type=cdrom
Label=CD-Rom1

[Drive M]
Path=/tmp
Type=hd
Label=Tmp Drive

[Drive N]
Path=${HOME}
Type=network
Label=Home
Filesystem=win95
```

Die Namengebung der Laufwerke geschieht mit [Drive x],die Laufwerksbuchstaben sollten allerdings den Laufwerken unter Windows entsprechen, Ausnahme sind hier hier M und N. Diese Laufwerke sind spezielle Verzeichnisse für wine. M ist das TEMP-Verzeichnis für wine, und mit N wird das Home-Verzeichnis unter Linux für die Windows-Programme sichtbar gemacht, und zwar als Netzwerk-Laufwerk (!). Mit *Path* werden den Laufwerksbuchstaben die entsprechenden Partitionen zugeordnet. *Type* legt den Typ der Partition fest, hd für Festplatte, cdrom für CD-Laufwerke,floppy für Disketten und network für Netzwerklaufwerke. *Label* legt die Bezeichnung für das Laufwerk fest, das Äquivalent zu dem label-Befehl unter MS-DOS. *Filesystem* legt den Typ des Dateisystems fest, es gibt die Möglichkeiten msdos,win95 und unix. Für VFAT, FAT32 und ext2fs sollte man win95 verwenden, für alte DOS-Partitionen msdos. In der zweiten Sektion der INI-Datei werden die globalen Einstellungen für wine gespeichert.

```
[wine]
Windows=c:\windows
System=c:\windows\system
Temp=m:\
Path=c:\windows;c:\windows\system;
SymbolTableFile=/usr/local/lib/wine.sym
```

In der [wine] Sektion werden die Pfade zur Windows-Installation, zum Temp-Verzeichnis und die von DOS bekannte Path-Variable gesetzt. Die SymbolTableFile ist nur für das Debuggen interessant.

```
[DllDefaults]
EXTRA_LD_LIBRARY_PATH=/usr/local/lib
DefaultLoadOrder = native,elfdll,so,builtin

[DllPairs]
kernel  = kernel32
gdi     = gdi32
user    = user32
commdlg = comdlg32
commctrl= comctl32
ver     = version
shell   = shell32
lzexpand= lz32
mmsystem= winmm
msvideo = msvfw32
winsock = wsock32

[DllOverrides]
kernel32, gdi32, user32 = builtin
kernel, gdi, user       = builtin
toolhelp                = builtin
comdlg32, commdlg       = elfdll,native, builtin
version, ver            = elfdll, builtin, native
shell32, shell          = builtin, native
```

```
lz32, lzexpand          = builtin, native
commctrl, comctl32      = native, builtin
wsock32, winsock        = builtin
advapi32, crtdll, ntdll = builtin, native
mpr, winspool           = builtin, native
ddraw, dinput, dsound   = builtin, native
winmm, mmsystem         = builtin
msvideo, msvfw32        = builtin, native
w32skrnl                = builtin
wnaspi32, wow32         = builtin
system, display, wprocs = builtin
wineps                  = builtin
```

Diese Abschnitte sind relativ neu und steuern das Laden der Windows-DLL's. Der Abschnitt [DllDefaults] enthält eine weitere Pfad-Varibale, die auf weitere Libraries verweist. Die DefaultLoadOrder bestimmt die Reihenfolge, in dem versucht wird die DLL's zu laden. Zur Zeit gibt es 4 verschiedene Möglichkeiten

builtin Die in WINE integrierten Funktionen zu nutzen

native Die original Windows-DLL's zu benutzen.

elfdll ELF[5]-gekapselte Windows-DLL's benutzen, momentan gibt es hier nur wenige Versionen

so ELF-Libraries benutzen

Diesen Eintrag sollte man nicht aendern. Die [DllPairs] spezifizieren, welche Win16 und Win32-DLL's zusammengehören.Diese Liste sollte man so übernehmen. wichtig ist hier vor allem, das sowohl Win16 als auch Win32-Version die gleichen Parameter benutzen. Der Abschnitt [DllOverrides] ist ebenfalls wichtig,hier kann man abweichtend von der DefaultLoadOrder die Reihefolge von besonders wichtigen DLL's festlegen. Folgende Ding sind hier besonders wichtig : kernel32, gdi32, user32,kernel, gdi, user, hier sollte immer die builtin-Version verwendet werden. Mit den anderen Versionen kann man experimentieren. Die Einstellung hier ist erprobt, allerdings kann jeder weiter experimentieren, wahrscheinlich wird sich hier bei neueren Versionen noch einiges tun.

```
[options]
AllocSystemColors=100

[fonts]
;Read documentation/fonts before adding aliases
Resolution = 96
Default = -adobe-times-

[Tweak.Layout]
WineLook=Win95

[serialports]
```

[5] Executable and Linkable Format - Format in dem unter Linux ausfuerhbare Dateien abgelegt sind

```
Com1=/dev/cua0

[parallelports]
Lpt1=/dev/lp0

[spooler]
LPT1:=|lpr


[spy]
Exclude=WM_SIZE;WM_TIME
```

In den [options] kann man die Farbtiefe für wine angeben, 100 Farben haben sich hier bewährt. Die [fonts]-Einstellung kann man auf der Default-Einstellung lassen,ebenso die [serialports], [parallelports],[spooler] und [spy]. Nur die Sektion [Tweak.Layout] sollte modifiziert werden, denn hier kann man das Look&Feel der Emulation festlegen. Zur Wahl stehen Win31,Win95 und Win98 Hinweis: Die Syntax der wine.conf wird öfters ergänzt und erweitert, es empfiehlt sich deshalb immer, einen Blick in die wine.ini im Source-Verzeichnis (siehe Kapitel 3) zu werfen, um auf dem Laufenden zu bleiben.

5 Ausführen von Programmen mit dem Emulator wine

Der Emulator wine wird mit

wine options program

gestartet. Es gibt eine Vielzahl von Optionen, die Man-Page von wine (man wine) enthält eine komplette Referenz für den Emulator. Nun zu den am häufigsten benötigten Optionen :

-winver Bestimmt, welche Windows-Version emuliert werden soll,zur Wahl stehen win31, win95, nt351 und nt40. Es ist ratsam, wine zu sagen, was er emulieren soll, da die automatische Bestimmung der Version des Windows-Programms nicht immer funktioniert.

-privatemap Eines der Vorteile des X-Window-Systems ist es, dass es jedem Programm eine eigene Farbpalette zur Verfügung stellen kann, wine kann diese Feature mit Hilfe dieser Option nutzen. Empfehlenswert, wenn die Windows-Programme Probleme mit der Farb-Darstellung haben.

-dll Mit dieser Option kann man die für einige Windows-DLL's[6] die Reihenfolge des Ladens verändern. Mit -dll comdlg32,commdlg=n,b werden zuerst die native Versionen benutzt und dann erst die builtin-Versionen. Allgemein sieht ein -dll so aus : $-dllDLL1 = n : DLL2, DLL3 = n, b, e$ Die Abkürzungen leiten sich aus den Bezeichnungen für die LAdereihenfolge ab. (siehe 4)

[6] Dynamic Link Library, eine Programmbibliothek, die zur Laufzeit vom Programm dazugeladen wird.

-managed Bewirkt, das die Windows-Fenster voll in die X-Window-Imgebung integriert werden. Statt dem Windows-Look, haben die Programme dann den Look des jeweiligen Window-Managers.

-language xx Bei Programmen, die die jeweiligen Landessprache unterstützen, kann diese Option zum Setzen des Ländercodes verwendet werden. Möglich sind En, Es, De, No, Fr, Fi, Da, Cz, Eo, It, Ko, Hu, Pl, Po, Sw, Ca, Ru.

-depth n Gibt wine die zu benutzende Farbtiefe an, hilft oft bei X-Servern, die hohe Farbtiefen verwenden.

Das auszuführende Programm kann entweder im DOS-Format oder im Linux-Format angegeben werden. Um z.B. Solitaire zu starten, kann man entweder $/dos/c/windows/sol.exe$ oder aber $c:\backslash\backslash windows\backslash\backslash sol.exe$ eingeben. Die doppelten Backslashes sind notwendig,damit die Shell sie richtig interpretiert. Wenn das Programm abstürzen sollte, dann wird der integrierte wine-Debugger aktiv, und man erhält eine komplette Analyse der Absturzursache. Den Debugger kann man mit *quit* verlassen.

6 WINE im INTERNET

Im Internet gibt es neben vielen privaten WWW-Seiten auch die offizielle WINE-Seite *www.winehq.com*, auf der sich eine Vielzahl von Informationen zu WINE befindet. Im Usenet gibt es eine eigene WINE-Newsgroup : *comp.emulators.ms-windows.wine* Dort wird auch bekannt gegeben, ob eine neue Version von WINE erschienen ist.

A Die wine-DLL's

A.1 Die Win16-DLL's

KERNEL	USER	GDI	WINDEBUG
AVIFILE	COMMDLG	COMPOBJ	DDEML
DISPDIB	KEYBOARD	COMM	LZEXPAND
MMSYSTEM	MOUSE	MSACM	MSVIDEO
OLE2CONV	OLE2DISP	OLE2NLS	OLE2PROX
OLE2THK	OLE2	OLECLI	OLESVR
RASAPI16	SHELL	SOUND	STORAGE
STRESS	TOOLHELP	TYPELIB	VER
W32SYS	WIN32S16	WIN87EM	WINASPI
WING	WINSOCK		

A.2 Die Win32-DLL's

ADVAPI32	AVIFIL32	COMCTL32	COMDLG32
CRTDLL	DCIMAN32	DDRAW	DINPUT
DPLAY	DPLAYX	DSOUND	GDI32
IMAGEHLP	IMM32	KERNEL32	LZ32
MPR	MSACM32	MSNET32	MSVFW32
NTDLL	OLE32	OLEAUT32	OLECLI32
OLEDLG	OLESVR32	PSAPI	RASAPI32
SHELL32	TAPI32	USER32	VERSION
W32SKRNL	WINMM	WINSPOOL	WNASPI32
WOW32	WSOCK32		

A.3 Welche Funktionen haben diese DLL's ?

Hier eine kurze Auflistung, welche Funktionen diese DLL's haben, soweit sie mir bekannt sind.

DLL	Version	Funktion
AVIFIL32	Win32	Support für AVI-Video-Dateien - 32 Bit Version
AVIFILE	Win16	Support für AVI-Video-Dateien
COMCTL32	Win32	Die Controls (Buttons etc.) von Windows95
COMDLG32	Win32	Die Dialogboxen von Windows 95
COMM	Win16	Unterstützung für die seriellen Ports
COMMDLG	Win16	Enthält die Standard Datei-öffnen, schliessen,etc. Dialogboxen
COMPOBJ	Win16	OLE 2.1 16/32 Bit-Schnittstelle
CRTDLL	Win32	Microsoft C-Runtime Library
DDEML	Win16	DDE Management
DDRAW	Win32	DirectDraw für direkten Grafikkartenzugriff
DINPUT	Win32	DirectInput für Joystick-Support
DISPDIB	Win16	VGA 256 Color DIB(Device Independent Bitmap)-Bibliothek
DPLAY	Win32	DirectPlay : Support für Multiplayer-Spiele über Netzwerk
DPLAYX	Win32	DirectPlay : Support für Multiplayer-Spiele über Netzwerk
DSOUND	Win32	DirectSound : Die Soundkartenunterstützung
GDI	Win16	Das Interface zur Grafikkarte
GDI32	Win32	Die Schnittstelle zur Grafikkarte in der 32 Bit-Version
KERNEL	Win16	Enthält die Systemroutinen von Windows
KERNEL32	Win32	Enthält die Systemroutinen von Windows 95
KEYBOARD	Win16	Die Tastaturunterstützung (naheliegend)
MMSYSTEM	Win16	System-APIs fr Multimedia
MOUSE	Win16	Mausunterstüzung
MPR	Win32	Netzwerk-Funktionen
MSACM	Win16	Tools zur Audiokomprimierung - 16 Bit
MSACM32	Win32	Tools zur Audiokomprimierung - 32 Bit
MSNET32	Win32	Die Netzwerk API
MSVFW32	Win32	Die MS-Video Funktionen für 32 Bit
MSVIDEO	Win16	Die alten MS-Video-Bibliotheken (erste Form von Video-Support für Windows)
OLE2	Win16	OLE2-Funktionen(Hauptbibliothek)
OLE2CONV	Win16	OLE2-Funktionen
OLE2DISP	Win16	OLE2-Funktionen
OLE2NLS	Win16	OLE2-Funktionen (Country-Codes)
OLE2PROX	Win16	OLE2-Funktionen
OLE2THK	Win16	OLE2-Funktionen
OLE32	Win32	OLE2 in der 32-Bit Version
OLEAUT32	Win32	OLE2 in der 32-Bit Version
OLECLI	Win16	OLE-Funktionen (Client-Funktionen)
OLECLI32	Win32	OLE2 in der 32-Bit Version(Client)

DLL	Version	Funktion
OLEDLG	Win32	OLE2 in der 32-Bit Version
OLESVR	Win16	OLE-Funktionen (Server-Funktionen)
OLESVR32	Win32	OLE2 in der 32-Bit Version(Server)
RASAPI16	Win16	16 Bit Dial-Up Netzwerk API für Modem/ISDN
RASAPI32	Win32	32 Bit Dial-Up Netzwerk API für Modem/ISDN
SOUND	Win16	Audio-Schnittstelle
STORAGE	Win16	OLE 2.3.1 16/32 Interoperability Bibliothek
TAPI32	Win32	Modem-Wähl-Support (Telephony API)
TYPELIB	Win16	OLE 2.1 16/32 Interoperability Bibliothek
USER	Win16	Die Schnittstelle für Benutzerprozesse
USER32	Win32	Die Schnittstelle für Benutzerprozesse - 32 Bit-Version
VER	Win16	Versionsprüfung und Dateiinstallation
WIN32S16	Win16	Win32S ermöglicht es einige Win32 Programme unter Win16 auszuführen
WIN87EM	Win16	Coprozessor-Emulation für Win16 (Aus der Zeit, als Coprozessoren noch nicht auf dem Hauptchip integriert waren
WINASPI	Win16	SCSI-Support-Schnittstelle
WING	Win16	Vorläufer von DirectX, praktisch nie benutzt
WINMM	Win32	System APIs for Multimedia
WINSOCK	Win16	TCP/IP Stack in der 16 Bit Version
WINSPOOL	Win32	Das Windows-Spool-System für Druckjobs
WNASPI32	Win32	Die Win32-SCSI-Schnittstelle
WSOCK32	Win32	TCP/IP Stack in der 32 Bit Version

B FAQ

Warum lässt sich WINE nicht compilieren ?

Es gibt Probleme WINE mit dem pgcc zu compilieren, ebenso machen die gcc-Versionen 2.7.2.1 und 2.7.2.2 Probleme mit dem Compilieren, das liegt wahrscheinlich daran, dass der Optimierer nicht mit dem WINE-Source zurecht kommt.

Wie kann ich DirectX testen ?

Dazu benötigt man Mut, Ausdauer und einen X-Server,der DGA-Support hat(alle X-Server von XFree86). Gefäehrlich ist allerdings, da man für die DirectX-Emulation momentan noch root-Rechte benötigt. Die Gefahr dabei ist, dass das Programm damit vollen Zugriff auf die Hardware erhält. Wenn die Emulation abstürtzt, wird die Grafikkarte nicht richtig resettet und man hat keinen Bildschirm-Output mehr. Deswegn ist es unbedingt empfehlenswert, den Test-Rechner von einem anderen Rechner aus herunterfahren zu können, oder vor Beginn des Testes einen shutdown -r -t sec einzugeben, damit der Rechner dann nach der abgelaufenen Zeit in Sekunden ordentlich heruntergefahren wird. Wenn alles glattgegangen ist, kann man den shutdown wieder abbrechen. Zumindestens StarCraft läuft mit voller Soundunterstützung sehr stabil. Mit anderen Spielen für Windows sieht es leider nicht so gut aus.

Welche Wine-Version ist empfehlenswert ?

Zumindestens nach Aussagen aus dem Usenet und eigenen Erfahrung ist 981211 recht stabil und kommt mit vielen Programmen zurecht, die 99er-Versionen haben alle Probleme mit der OLE-Initialisierung, weil hier gerade viel Redesign stattfindet. Das ist auch ein bekanntes Problem. Wer also an der Spitze der Entwicklung stehen will, kann sich mit akutell 990704 vergnügen, wer sicher gehen will, soll 981211 benutzen. Folgendes sollte man aber beachten, die alten Versionen (bis 9903) haben ein etwas anderes Format der Konfigurationsdatei als die neueren Versionen, auch die Kommandozeilenoptionen haben sich leicht geaendert.

Wieso findet das Programm das CD-ROM nicht ?

Die Lösung ist recht einfach, das CD-ROM muss zuerst unter Linux gemountet werden.

C Schlusswort

Ich hoffe, diese BIC beantwortet viele Fragen zu WINE. Die BIC erhebt weder Anspruch auf Vollstndigkeit noch Fehlerfreiheit. Wenn diese BIC dazu führt, das sich noch weitere Leute mit WINE beschäftigen und vielleicht auch noch zu dieser BIC beitragen, dann hat sie ihr Ziel erreicht. Für Verbesserungsvorschläge,Fehlerkorrekturen, Erweiterungen usw. bitte EMAIL an :

ubg9@rz.uni-karlsruhe.de