Seminar Linux Grundlagen
23.09. 1999

Christian Schneider
Michael Karch

Seminarunterlagen

Vorstellung von Linux

Linux ist ein Unix - Betriebssystem. Der wichtigste Unterschied gegenüber anderen Unix-Systemen besteht darin, daß Linux frei kopiert werden darf. Einschränkungen in der Funktionalität gibt es dagegen kaum. Linux ist in vielerlei Beziehung vollständiger als so manches kommerzielle Unix-System. Es unterstützt eine größere Palette von Hardware - Komponenten und enthält in vielen Bereichen effizientere Codes.

Was ist Linux ?

Genaugenommen bezeichnet der Begriff „ Linux“ nur den Kernel: Der Kernel ist der innerste Teil (Kern) eines Betriebssystems mit ganz elementaren Funktionen wie Speicherverwaltung, Prozeßverwaltung und der Steuerung der Hardware.

Distributionen

Einige Firmen haben es sich zur Aufgabe gemacht Linux zu fördern. Diese Firmen arbeiten neben vielen freiwilligen Entwicklern und Programmierern an der Weiterentwicklung von Linux indem sie Programme, Treiber und weitere Pakete entwickeln und frei (nicht mit kostenlos zu verwechseln) zur Verfügung stellen (z.B.: YAST, SAX, KDE, Gnome, Samba etc.).

Caldera Open Linux, Debian GNU/Linux, DLD, RedHat, Slackware, SuSE sind Distributionen die als Komplettpaket mit Programmen, Paketen und Installationstools etc. erhältlich sind, jede Distribution ist unterschiedlich und unterscheidet sich abgesehen vom Kernel (Versionsunterschiede kommen vor) grundsätzlich in den beigefügten Programmen, Installationstools, Organisation des Rechnerstarts (Init-V-Boot-Prozesses), Dokumentation, Support etc. Die Frage, welche Distribution die beste ist, welche wem zu empfehlen sei etc., artet leicht zu einem Glaubenskrieg aus. Kriterien für die Auswahl einer Distribution sind die Aktualität ihrer Komponenten, die Qualität der Installations- und Konfigurationstools, der angebotene Support, mitgelieferte Handbücher etc.

Einsatzgebiete (Vor- und Nachteile)

Einsatzgebiete:

Linux kann auf einfach ausgestatteten PCs betrieben werden und unterstützt eine Menge von Servicen und Diensten standardmäßig, ohne Zukauf von zusätzlich teurer Software, den bei Linux ist die meiste Software schon kostenfrei dabei. Die folgende Liste gibt einen guten Überblick über die wichtigsten Einsatzgebiete :

- Web Server
- Router
- News Server
- FTP Server
- File Server
- Workstation
- Access Server
- Proxy Server
- Datenbank - Server
- Mail Server

Vorteile:

- Linux darf frei kopiert werden
- Multitasking (gleichzeitige Abarbeitung mehrerer Prozesse)
- Multiuser-Betrieb (gleichzeitige Nutzung durch mehrere Anwender)
- Minimum Anforderung: 386 mit 8MB RAM und 300 MB Speicherplatz
- Linux unterstützt nicht nur INTEL Prozessoren, sondern auch den DEC-Alpha-Prozessor, Sun Sparc, Mips, Motorola etc.
- Linux verwendet ein eigenes Dateisystem (255 Zeichen, Dateien bis zu 2 Gbyte und Dateisysteme bis zu 4 Tbyte groß)
- Linux kann auf viele fremde Dateisysteme zugreifen: DOS, Windows 95, OS/2, Minix etc.
- Stabiler (stürzt nicht so oft ab wie andere Betriebssysteme z.B.: Windows)
- Frei kompilierbarer Kernel

Nachteile:

- Für „verwöhnte“ Windows Anwender gewöhnungsbedürftig ist, da meist im Textmodus vorkommend
- professionelle Software ist noch rar
- die graphische Bedienung ist noch nicht ganz so ausgereift wie bei Windows

Empfohlene Minimale Hardwareausstattung :

Server (nur Textmodus) : 486-100 MHz, 32MB Ram, 0,5 - 1GB HD
Workstation (X11) : P133 64MB Ram, 1,5-2 GB HD

Dokumentation, Hilfesysteme

Die meisten Einstellungen in einem Linux System sind noch nicht grafisch durchführbar und daher nicht einfach intuitiv erlernbar. Glücklicherweise wird aber eine riesige Fülle an Hilfen bei einem Linux System mitgeliefert.

Zum einen gibt es verschiedene Hilfesysteme, die von der Kommandozeile bedienbar sind :

man

Das ursprüngliche Hilfesystem von Unix. Über die man pages bekommt man Hilfe zu Programmen, Konfigurationsdateien und zur Programmierung.

Hilfe zum man System selbst bekommt man mit dem Kommando **man man**.

Um die verschiedenen Arten von Hilfetexten zu unterscheiden sind die man pages in Sections unterteilt :

1 Executable programs or shell commands
2 System calls (functions provided by the kernel)
3 Library calls (functions within system libraries)
4 Special files (usually found in /dev)
5 File formats and conventions eg /etc/passwd
6 Games
7 Macro packages and conventions eg man(7), groff(7).
8 System administration commands (usually only for root)
9 Kernel routines [Non standard]

Die einfache Aufrufsyntax von man ist man <Schlüsselwort>. Dann benutzt die erste Section die das Schlüsselwort enthält.

Wenn das Schlüsselwort in mehreren Section vorkommt kann das Hilfesystem aber auch mit **man <Section-Nummer> <Schlüsselwort>** aufgerufen werden um die Section direkt anzuwählen.

info

Das info System ist der Nachfolger von man. Dieses System erlaubt ähnlich wie html Hyperlinks und unterteilt die Hilfe zu einem Befehl in mehrere Seiten.

Einige Programme wie z.B. tar werden nur in einer Info Seite dokumentiert.

Der Aufruf ist **info <Schlüsselwort>.**

andere Hilfen

Außer den Hilfesystemen gibt es noch eine große Menge von Hilfetexten im Verzeichnis /usr/doc. Diese zeigt man am besten mit dem Kommando less

<Dateiname> an, da dieses auch direkt komprimierte Dateien (Endung .gz) anzeigen kann.

Das Verzeichnis /usr/doc gliedert sich in verschiedene Unterverzeichnisse :

/usr/doc/howto	Howtos sind praxisorientierte Hilfen zum Einrichten verschiedener Dienste und Systemfunktionen. Im Gegensatz zu den man pages, die oft nur die Syntax beschreiben findet man hier detaillierte Anleitungen, wie die Einrichtung und der Test der Pakete durchgeführt wird
/usr/doc/packages	Hier speichert S.u.S.E linux die Dokumentationen, die mit den Paketen mitgeliefert werden. Diese sind zwar oft nicht ganz so gut geschrieben wie die Howtos aber man findet oft Beispiele für Konfigurationsdateien, die seht hilfreich sind.
/usr/doc/susehilf	Einige Hilfen von S.u.S.E im Html Format

Verzeichnisstruktur

Die Dateisystemstruktur von Linux ist recht komplex und für den Anfänger nicht leicht durchschaubar. Dazu kommt noch, daß sich diese Struktur bei den verschiedenen Unix Varianten an einigen Stellen unterscheidet. Die folgende Tabelle bezieht sich auf S.u.S.E Linux. Sie ist aber in weiten Teilen auch für andere Systeme gültig.

/bin	Grundlegende Programme (z.B. cp,mv)
/dev	Spezielle Dateien, die Zugriff auf Geräte ermöglichen
/etc	Konfigurationsdateien
/home	Benutzerverzeichnisse
/lib	Dynamische Bibliotheken (wie dlls in Windows)
/lost+found	Dateien, die bei einem fscheck wegen fehlender Verknüpfungen nicht mehr einem Verzeichnis zugeordnet werden konnten
/mnt	Üblicher Ort um zusätzliche Dateisysteme zu mounten
/opt	Sinnvoller Platz, um selbstinstallierte Pakete abzulegen
/proc	Dieses Verzeichnis enthält keine richtigen Dateien, sondern ermöglicht Zugriff auf verschiedene Statusinformationen
/root	Benutzerverzeichnis des Systemadministrators
/sbin	Programme, die normalerweise nur der Systemadministrator benötigt (z.B. lilo, ifconfig)
/sbin/init.d	Init Skripten (zum Starten und beenden der Serverdienste)
/sbin/init.d/rc2.d	Enthält Links für die im Textmodus zu startenden Dienste
/sbin/init.d/rc3.d	Enthält Links für die im Grafikmodus zu startenden Dienste
/tmp	Temporäre Dateien
/usr	Zusätzliche Programme, Bibliotheken und Konfigurationsdateien. Diese sind in einem getrennten Verzeichnis untergebracht, damit sie auch von einem Server gemountet oder auf eine eigene Partition gelegt werden können.
/usr/doc	Wenn die Dokumentation installiert wurde befinden sich hier sehr viele Hilfen zu den einzelnen Paketen und zur Konfiguration des Systems (genaueres dazu im nächsten Kapitel)
/usr/local	Wie /opt
/var	Daten und Logdateien des Systems (keine Benutzerdaten)

/var/log	Logdateien
/var/spool	Durchgangsverzeichnis für Mail, Drucker, Fax ...
/var/spool/mail	Ausgelieferte Mails der Benutzer
/var/spool/mqueue	Noch nicht ausgelieferte Mails
/var/run	Prozessnummern der laufenden Dienste (Werden von den init Skripten benutzt, um diese wieder zu beenden)

Bedienung der Shells

Shells sind die Bedienoberfläche von Linux im Textmodus. Außerdem ermöglichen sie die Programmierung von sehr mächtigen Skripten.

Die einfachste shell ist sh. Sie ist nicht besonders angenehm zu bedienen und bietet auch in der Programmierung keine Finessen aber sie ist auf jedem System vorhanden. Viele Skripten benutzen daher diese Shell.

Meine Lieblingsshell ist die bash shell (bourne again shell), die eine Weiterentwicklung von sh darstellt.

Die bash shell bietet einige sehr komfortable Erweiterungen zur Bedienung.

- Mit den Cursortasten links und rechts kann man innerhalb eines Befehls editieren und so einen Eingabefehler beheben. Standardmäßig ist hierbei der insert modus aktiviert, so das man angenehm auch neuen Text einfügen kann.
- Die Cursortasten hoch und runter erlauben wie Doskey in Windows alte Befehle wieder aufzurufen.
- Die wohl angenehmste Eigenschaft der bash ist die Tab-Ergänzung. Wenn man einen Teil eines Dateinamens getippt hat kann man diesen durch einmaliges drücken von Tab vervollständigen lassen, sofern er schon eindeutig ist. Wenn es mehrere Möglichkeiten zur Ergänzung gibt, so kann man diese durch einen zweiten Druck von Tab anzeigen lassen. Besonders angenehm ist diese Funktion wenn man einen Programmnamen nur teilweise kennt.

Wenn man eine Datei auf der ganzen Platte suchen möchte so gibt es dazu zwei Wege. Zum einen kann man **find . –name <Dateiname> .** benutzen. Diese Funktion durchsucht das Dateisystem rekursiv ab der momentanen Position.

Schneller geht das ganze über den Befehl **locate <Dateiname>**. Diese arbeitet über eine Datenbank und liefert daher sehr schnell die Liste aller Dateien, die dem Suchbegriff entsprechen. Allerdings ist locate nicht auf jedem System installiert und auch nicht immer auf dem neuesten Stand, da die Datenbank nur jeden Tag einmal aktualisiert wird. Wenn man gerade ein Paket neu installiert hat und darin eine Datei sucht muß man vor locate noch updatedb aufrufen, um die Datenbank zu aktualisieren. (kann aber eine Weile dauern.

Kommandozeilenbefehle

Unix Befehl	Dos Befehl	Beschreibung
man		Hilfe zu einem Befehl anzeigen
exit	exit	Shell verlassen
ls -al	dir	Verzeichnis ansehen
ls	dir /w	Verzeichnis in Kurzform ansehen
cd	cd	Verzeichnis wechseln
pwd		Aktuellen Pfad ansehen
cp	copy	Dateien kopieren
mv	move	Dateien verschieben
rm	del	Dateien löschen (vorsicht, es gibt Standardmaessig keine Sicherheitsabfrage!)
mkdir	md	Verzeichnis anlegen
rmdir	rd	Leeres Verzeichnis löschen
chmod		Rechte setzen
chown		Besitzer setzen
joe	edit	Texteditor
vi	edit	Texteditor (kompliziert, aber funktioniert mit jedem Terminal)
export	set	Umgebungsvariablen setzen
passwd		ändern des Benutzerpassworts
su		Auf andere userid wechseln (nur als root)
telnet	telnet	
ftp	ftp	
more, less	type	Dateien anzeigen
ps		Anzeige der laufenden Prozess
kill		Beenden eines Prozesses
grep		Anzeige aller Zeilen, die einen bestimmten Suchbegriff enthalten
ping	ping	Überprüfen der Verbindung zu einer IP Adresse
traceroute	tracert	Anzeige des Wegsm den ein IP Paket vom lokalen Rechner zum Ziel nimmt
nslookup		Anzeigen der IP Adresse zu einem Rechnernamen oder umgekehrt
locate		Schnelles Datenbankbasiertes suchen von Dateien (auf Sinix nicht verfügbar)
updatedb		Update der locate Datenbank
find		Suchen von Dateien
which bzw. type -p		Zeigt den Pfad zum eingegebenen ausführbaren Programm an
ln		Erzeugt einen Link auf eine Datei

Terminaltypen

Solange man direkt vor einer Linux Console sitzt oder über xterm auf eine Linux System zugreift wird man sich selten um den Terminaltyp kümmern müssen. Wenn man allerdings über telnet auf eine andere Unix Variante zugreift erzeugen die Standardeinstellungen meist nur Schrott. Dies liegt daran, daß andere Systeme normalerweise mit dem von Linux verwendeten Terminal wenig anfangen können.

Um auf solchen Systemen einigermaßen angenehm zu arbeiten muß man folgende Kommandos verwenden :

export TERM=vt100	Setzt den Terminaltyp auf ein relativ einfaches Terminal, das auf jedem System bekannt ist
setenv TERM=vt100	Für einige shells statt der obigen Variante zu verwenden
stty intr ^C	Setzt Break auf Ctrl C. (Dieses ist oft standardmäßig auf Backspace gesetzt, was bewirkt, daß man keine Eingaben löschen kann
stty erase ^?	Setzt Backspace auf die richtige Sequenz

Diese Kommandos kann man auch in eine Datei term schreiben, diese mit chmod u+x term ausführbar machen und dann per **. ./term** aufrufen. Der einzelne Punkt ist wichtig, da sonst die Änderungen in der Variable TERM nicht auf die aufrufende Shell übertragen werden.

Editoren („joe“ und „vi“)

Bei Linux Distributionen sind meist Editoren zur Bearbeitung von Dateien enthalten. Wir wollen zwei verschieden Editoren kurz vorstellen.
Zum einen den sehr einfachen Texteditor joe zum anderen den recht komplizierten aber sehr kompakten Editor „vi“.

joe

Mit man joe erhalten Sie einen umfangreichen Manual-Text zu joe. Dieser Text beschreibt alle weiteren Kommandos. Um den Editor zu nutzen muß man einfach den Editornamen und nachfolgend den Dateinamen eingeben (Achtung: nicht vergessen den Pfad anzugeben). Beispiel:

joe /etc/rc.config (dabei ist etc der Pfad und rc.config die Datei)

Die wichtigsten Kommandos für joe:

STRG+K, H	blendet das Hilfefenster ein/aus
STRG+K, E	lädt eine neue Datei
STRG+K, D	speichert die Datei
STRG+Y	löscht eine Zeile
STRG+Shift+_	Undo (Löschen rückgängig machen)
STRG+C	beendet joe (mit Rücksprache zum Speichern)
STRG+K, X	beendet joe mit automatischer Speicherung (ohne Rückfrage)

vi

vim und elvis sind zwei vi- kompatible Editoren. Der Original-vi ist aus urheberrechtlichen Gründen nicht Teil von Linux. Das Kommando vi kann aber zumeist dennoch ausgeführt werden und bewirkt dann automatisch den Start von vim oder elvis.
Wir erwähnen vi nur, da viele Anfänger wenn Sie einmal vi gestartet haben nur noch mit einem Restart des Rechners den Editor beenden können. Wenn Sie noch nie mit vi gearbeitet haben, sollten Sie es gar nicht erst probieren – die Bedienung ist (höflich formuliert) unkonventionell. Der einzig echte Vorteil des vi gegenüber anderen Editoren besteht darin, daß der Editor sehr kompakt ist und eventuell auch dann zur Verfügung steht, wenn kein anderer Editor läuft.
Bei vi unterscheidet man zwischen Textmodus (Input) und Kommandomodus, die Beschreibung spricht schon für sich. Im Textmodus kann der Text bearbeitet werden und im Kommandomodus müssen Befehle wie speichern und exit eingegeben werden. Mit der Taste ESC wechselt man aus dem Textmodus in den Kommandomodus.

Kommandos :

i	Insert Modus (ermöglicht, Text einzugeben)
a	Append Modus (wie insert, aber funktioniert auch am Ende einer Zeile)
x	Löscht einen Buchstaben
dd	Löscht eine Zeile
q	beenden ohne speichern
q!	beenden ohne speichern. Funktioniert auch wenn am Dokument Änderungen vorgenommen wurden
wq	beenden mit speichern
wq!	beenden mit speichern. Funktioniert auch wenn Datei read only ist (als root)

Zugriffsrechte für Dateien

Im Gegensatz zu Windows ist Unix ein Multi User Betriebssystem. Da auf einer Maschine normalerweise mehrere Personen Login-Berechtigungen haben wird unter Unix sehr großen Wert auf den Schutz von Dateien gelegt.

Die Details zu Benutzern werden in Unix in der Datei /etc/passwd gespeichert, die Gruppen werden in /etc/group definiert.

Übrigens keine Sorge, trotz des Namens stehen in /etc/passwd keine Paßwörter. Diese sind z.B. bei Linux in /etc/shadow zu finden. Diese Datei ist aber 1. nur für root lesbar und 2. enthält sie die Paßwörter in einer verschlüsselten Form.

Grundsätzlich gehört jede Datei einem bestimmten Benutzer und sie ist auch einer bestimmten Benutzergruppe zugeordnet.

Um diese Daten anzuzeigen benutzt man ls -al.

Hier ein kleines Beispiel :

Rechte	Verzeichnis-einträge	Besitzer	Gruppe	Größe	Datum	Name
drwxr-xr-x	7	chris	users	1024	Mar 10 21:06	.
drwxr-xr-x	6	root	root	1024	Feb 22 03:30	..
-rw-r-----	1	chris	users	1433	Mar 4 02:16	datei1
-rw-rw-r--	1	tom	users	3824	Mar 4 02:16	datei2

datei1 gehört Benutzer chris und ist der Gruppe users zugeordnet.
datei2 gehört Benutzer tom und ist ebenfalls der Gruppe users zugeordnet.

Den Besitzer einer Datei ändert man mit chown <Dateiname>, die Gruppe mit chgrp <Dateiname>.

Etwas verwirrend an obiger Verzeichnisanzeige ist vielleicht die erste Spalte, in der die Rechte angezeigt werden. Hier eine genaue Beschreibung dieser Spalte.

Grundtyp	Rechte für Besitzer der Datei	Rechte für alle Benutzer, die der gleichen Gruppe wie die Datei angehören	Rechte für alle anderen Benutzer
d	rwx	r-x	r-x

Es existieren folgende Bezeichner für Grundtypen :

-	normale Datei
d	Verzeichnis
s	setuid Datei (wird mit den Rechten des Besitzers ausgeführt)
l	Link

Die Zugriffsrechte können für den Besitzer, die Mitglieder der Gruppe und alle anderen Benutzer getrennt definiert werden.

Im Fall einer Datei bedeuten die drei Zeichen :

r	Leserecht
w	Schreibrecht
x	Ausführungsrecht

Im Fall eines Verzeichnisses :

r	Verzeichnis anzeigen
w	Dateien anlegen und löschen
x	Recht in das Verzeichnis zu wechseln

Beispiele :

datei1 ist für Benutzer Chris lesbar und schreibbar, für die Mitglieder der Gruppe users Lesbar. Alle andren Benutzer haben keinen Zugriff auf die Datei.

datei2 ist für Benutzer Tom lesbar und schreibbar, für die Mitglieder der Gruppe users lesbar und schreibbar. Alle anderen Benutzer dürfen die Datei nur lesen.

Bedienung von chmod

Der Befehl chmod dient dazu, die Dateirechte zu verändern.

Es gibt zwei verschiedene Arten, chmod aufzurufen :

1. mit numerischer Angabe der Rechte :

Beispiel :

chmod 640 datei1 (setzt Datei1 auf lesen+schreiben für Besitzer, lesen für Gruppe und keine Rechte für alle anderen Benutzer)

Die drei Ziffern repräsentieren die Benutzerrechte, Gruppenrechte und die Rechte für andere Benutzer

Jede Ziffer setzt sich aus folgenden Summanden zusammen :

4	Leserecht
2	Schreibrecht
1	Ausführungsrecht

Beispiel :

2+4 = 6 (d.h. lesen und schreiben).
1+2+4 = 7 (d.h. lesen, schreiben und ausführen)
0 bedeutet keine Rechte

2. durch Angabe der zu ändernden Benutzerart und den zu verändernden Rechten

Beispiel :

chmod u+rw datei1 (Gibt dem Besitzer Lese- und Schreibrechte)
chmod g-w datei1 (Nimmt der Gruppe die Schreibrechte)
chmod a=rw datei1 (Setzt Lese- und Schreibrechte für Benutzer, Gruppe und Andere)

Der erste Buchstabe bezeichnet für wen die Rechte verändert werden sollen. Möglich sind :

u	Benutzer
g	Gruppe
o	Andere
a	Alle

Danach folgt die Art der Änderung :

+	Rechte hinzufügen
-	Rechte entfernen
=	Auf diese Rechte setzen

Danach folgt eine Aufzählung, welche Rechte verändert werden sollen.

r	Leserecht
w	Schreibrecht
x	Ausführungsrecht

Pipes, Ein-/Ausgabeumleitung

Eigentlich sind Pipes und Umleitungen eher etwas für ein fortgeschrittenes Linux Seminar. Da aber die Syntax zum Teil aus Dos bekannt ist und sich oft als sehr nützlich erweist beschreiben wir hier die nötigen Grundlagen.

Es gibt folgende Arten der Umleitungen

Programm > Dateiname	**Ausgabeumleitung** Alles was das Programm eigentlich auf stdout (normalerweise der Bildschirm) schreiben wollte wird in die Datei umgeleitet
Programm < Dateiname	**Eingabeumleitung** Das Programm erhält seine Eingabe (stdin) statt wie üblich von der Tastatur aus der angegebenen Datei
Programm 2> Dateiname	**Umleitung der Fehlerausgabe** Alles was das Programm an stderr (normalerweise ebenfalls der Bildschirm sendet geht in die genannte Datei
Programm_1 | Programm_2	**Pipe** Die Standardausgabe von Programm 1 wird auf die Standardeingabe von Programm 2 geleitet

Nach dieser ganzen Theorie mal einige sinnvolle Beispiele für Umleitungen :

cat /var/log/messages | more	Listet die Datei /var/log/messages aber ermöglicht dem Benutzer die Seiten durchzublättern
ls > dir.txt	Schreibt die Anzeige des aktuellen Verzeichnisses in die Datei dir.txt
ftp < kommandos.txt	Steuert den ftp Client mit den Kommandos, die in der Datei kommandos.txt stehen
tar –cz /texte | mail chris	Packt die Dateien im Verzeichnis Texte in ein Tar Archiv und sendet sie an den Benutzer chris

Umgang mit Wechselmedien

Im Gegensatz zu Windows ist der Umgang mit CD-ROM und Disketten anfänglich sehr kompliziert und kostet einige Zeit bis man diese Medien gelernt hat zu nutzen. Disketten und CD‘s müssen unter Linux „gemountet“ werden, d.h. das Medium wird in die Verzeichnisstruktur eingebunden. Es gibt bei Linux keine Laufwerksbuchstaben (A:, D:) wie unter Windows.

Die Datei /etc/fstab erleichtert uns die Arbeit

Device	Mountpoint	Typ	Optionen		
/dev/hda1	swap	swap	defaults	0	0
/dev/hda2	/	ext2	defaults	1	1
/dev/hdb	**/cdrom**	**iso9660**	**ro,noauto,user**	**0**	**0**
/dev/fd0	**/floppy**	**auto**	**noauto,user**	**0**	**0**
proc	/proc	proc	defaults	0	0

Die zwei fett markierten Einträge sind für das mounten von CD und Diskette verantwortlich, vor diesen beiden Einträge darf keine Raute (#) stehen. Durch diesen Eintrag in der fstab reicht nun der Befehl:

mount /cdrom/ für CD-ROM
mount /floppy/ für die Diskette

um die Medien zu mounten.

Weitere Infos erhaltet ihr mit **man mount** und **man fstab**.

Nun werdet ihr bemerken, daß eine CD nachdem Sie in die Verzeichnisstruktur eingebunden wurde (gemountet) sich nicht mehr aus dem Laufwerk entfernen läßt. Das Laufwerk ist „abgeschlossen“ (verriegelt). Warum, hättet ihr lieber einen Blue Screen wie bei Windows ???????
Der Befehl „**umount“** entfernt die CD oder Floppy wieder aus der Verzeichnisstruktur, so das ihr die CD oder Diskette wieder entnehmen könnt.

Beispiel :

umount /cdrom/
umount /floppy/

Andere mögliche devices für CD-rom Laufwerke und Festplatten sind :

/dev/hda	Primärer IDE Controler Master
/dev/hdb	Primärer IDE Controler Slave
/dev/hdc	Sekundärer IDE Controler Master
/dev/hdd	Sekundärer IDE Controler Slave
/dev/sda	SCSI Festplatte mit niedrigster ID
/dev/sdb	2. SCSI Festplatte
/dev/scd0	Erstes SCSI cdrom

Die Nummer hinter dem Device bezeichnen die Partition so z.B. hda1 für die erste Partition.

Wenn ihr euch nicht sicher seid könnt ihr euch die erkannten Platten und Partitionen mit cat /var/log/boot.msg ansehen.