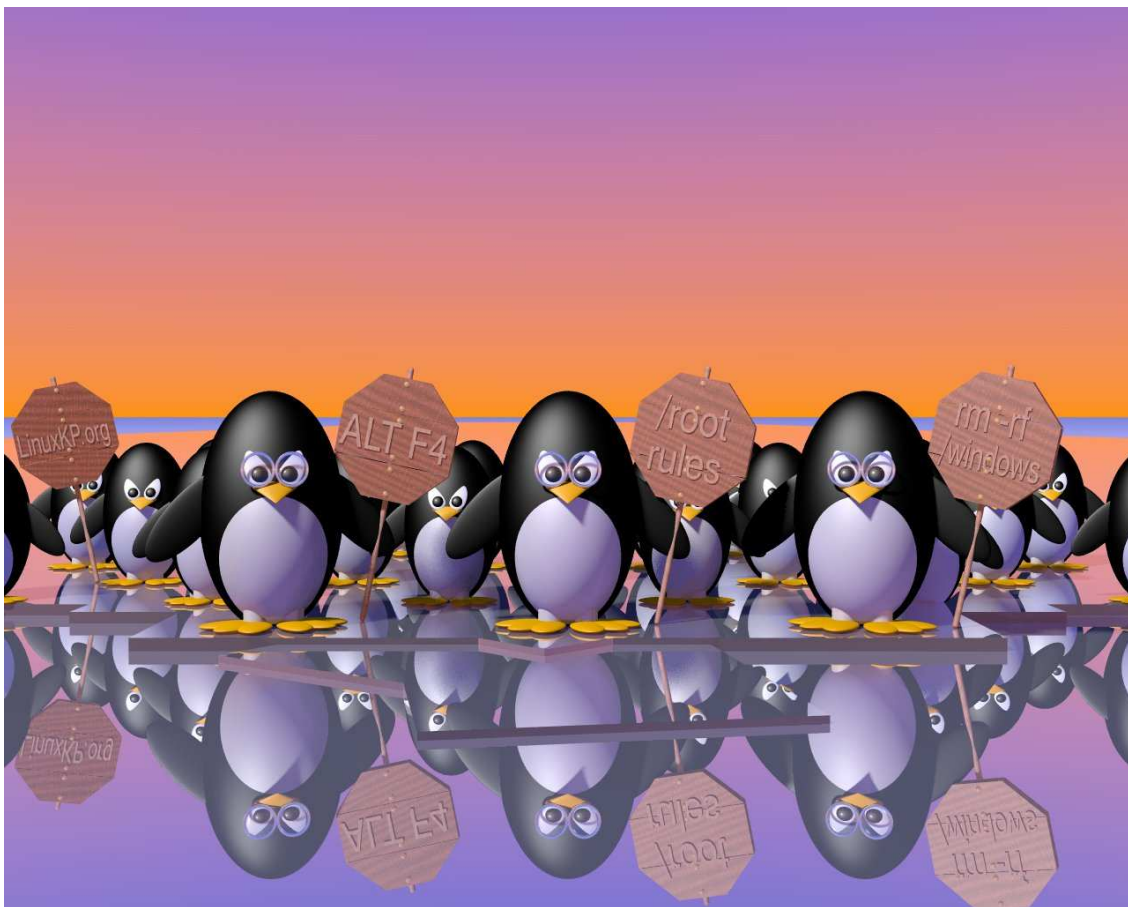


Linux Knowledge Portal **Linux für Umsteiger**

<http://LinuxKP.org>



Autoren: Stephan Barth, Kurt Garloff, Jana Jaeger, Frank Rennemann, Martin Sommer

Zusammengestellt von Frank Rennemann, Version 1.0

18. Oktober 2004

Copyright © 2004 Frank Rennemann.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with the Invariant Section *GNU Free Documentation License*, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled *GNU Free Documentation License*.

Waren- und Markennamen werden ohne Rücksicht auf die Gewährleistung der freien Verwendbarkeit benutzt. Fast alle Hardware- und Softwarebezeichnungen, die in diesem Dokument erwähnt werden, sind gleichzeitig eingetragene Waren- oder Markenzeichen oder sollten als solche betrachtet werden.

Linux ist eingetragenes Warenzeichen von Linus Torvalds.

Inhaltsverzeichnis

1	Ein paar Worte vorweg...	5
1.1	Der Erstkontakt	5
1.2	Jetzt geht's los	6
1.3	Herkunft der Texte	9
1.4	Motivationsgründe	9
2	Linux ist anders	11
2.1	Das Konzept: Multiuser und Multitasking	11
2.2	Programmphilosophie	13
3	Leitfaden für Umsteigewillige	17
3.1	Die Hardware	17
3.2	Die Software	18
3.3	Welches Linux soll ich nehmen?	19
3.4	Das planvolle Vorgehen	19
3.5	...und die alten Daten?	20
3.6	Man nehme...sich Zeit	22
3.7	In medias res	22
3.8	Allgemeines zur Installation	23
3.9	Linux zum Ausprobieren	25
3.10	Was mache ich womit?	25
3.11	Wer hilft mir, wenn was schiefgeht?	26
3.12	Links	26
4	Kleine Baumkunde für Jungpinguine	27
4.1	Linux ist anders - Dateisysteme und Partitionen	27
4.2	Wo liegt was? - Der Dateibaum unter Linux	29
4.3	und weiter...	31
5	Dateisysteme unter Linux	33
5.1	Was genau tut ein Dateisystem?	33
5.2	Welche Dateisysteme gibt es?	33
5.3	Zwei aus X	34

6	Innenansichten eines Linuxsystems	39
6.1	lspci	39
6.2	grep	39
6.3	RPM	40
6.4	Informationen über nicht installierte Pakete	42
6.5	Weitere Hilfen	43
7	Wichtige Befehle für Umsteiger	45
7.1	Umgang mit Dateien und Verzeichnissen	46
7.2	Kopieren zwischen Workstation und Server	49
7.3	Eigentümer und Rechte vergeben	49
7.4	Prozesse beenden	51
7.5	Automatische Namensergänzung und History	52
8	Windowmanager	54
8.1	Was tut ein Windowmanager?	54
8.2	Kurzportraits	55
8.3	Wo hört ein Windowmanager auf?	58
9	With a little help from my friends ...	60
9.1	Hilfe zur Selbsthilfe?	60
9.2	Hardware-Informationen	61
9.3	Hilfe vom Rest der Welt	64
9.4	... in eigener Sache	67
A	GNU Free Documentation License	I

Kapitel 1

Ein paar Worte vorweg...

Wann immer sich ein paar IT'ler mit Pinguin-Anstecker am Revers oder Tux-lastigem Werbe-T-Shirt treffen und Zeit haben, in der Vergangenheit zu schwelgen, in der sowieso alles viel cooler, interessanter, anspruchsvoller etc. gewesen ist, wird einer anfangen, davon zu schwärmen, was für tolle Dinge er mit seinen ersten Rechnern angestellt hat. Ähnlich wie Geschichten aus der Schule oder dem Studium werden auch die IT-Geschichten immer besser, je länger sie zurückliegen¹.

Damals, als man Kernel noch monolithisch baute und vor allem selbst kompilierte, als fertige Kernelmodule für Modem und Netzwerkkarte noch in der fernen Zukunft lagen und niemand daran dachte, Sound unter Linux zu hören, denn wozu braucht ein Server schon eine Soundkarte? Damals, als KDE und Gnome noch Überschriften in den Albträumen ihrer späteren Entwickler waren und fvwm2 das Nonplusultra der X-Window-Benutzer war.

Was, Sie wissen nicht, was ein Kernel ist? Oder ein Kernelmodul? Und warum man sowas kompilieren sollte? Was ist überhaupt kompilieren? Keine Angst, genau zur Beantwortung solcher und ähnlicher Fragen ist dieses Buch von uns geschrieben worden.

Um Ihnen zu zeigen, daß auch fortgeschrittene Linuxer klein angefangen haben, wird dieses Vorwort um eine Schliderung meiner ersten Linux-Installation bereichert...

1.1 Der Erstkontakt

Noch vor meinem ersten eigenen DOS-PC hatte ich mal von einer Linux-Installationsparty im Uni-Rechenzentrum in Hannover gehört. Leider gab es das nicht für ATARI's und einen PC hatte ich damals noch nicht. Als ich dann einen hatte, war die Festplatte so klein, das außer DOS und Windows nichts mehr drauf passte. Also

¹vor allem vergißt man die zahlreichen Unzulänglichkeiten, mit denen man an den alten Rechnern zu kämpfen hatte.

hie es warten, bis Geld fr eine zweite Festplatte da war. Dann wurde flugs ein Linux-Buch gekauft (der Kofler[5], sozusagen ein deutschsprachiges Standardwerk fr Linux), denn da war eine Slackware-Distribution drauf, Version 2.1 mit Kernel 1.1.86².

Das Installieren des Grundsystems gestaltete sich recht ertrglich, aber von Fenstern keine Spur. Also schlau lesen, ach ja, Xfree86, das ist dann also X-Windows³, oder wie. Was, das mu an jede Hardware einzeln angepasst werden? Irgendwo zwischen diversen Konfigurationsversuchen waren dann die Semesterferien zu Ende und ich gab erstmal wieder auf. Bis dahin war es mir trotz mehrerer Anlufe nicht gelungen, einen X-Server zum Laufen zu bringen und auch mein Drucker hatte keine Zeile von sich gegeben. Also Festplatte neu formatieren und den Platz fr Spiele nutzen. Das verdrngte zwar meinen Frust, gescheitert zu sein, aber innerlich wurmte es mich doch. Da gab es ein absturzsicheres Betriebssystem, ein richtiges Betriebssystem, kein Disketten-Operations-System. Das ging sogar mit den Ressourcen viel effektiver um als DOS und man konnte, anders als in Windows, mehrere verschiedene Konfigurationen und Umgebungen schaffen. Und T_EX war schon drin. Also wurde der Plan nicht aufgegeben, nur vorlufig vertagt. Ganz ohne Modem machte Linux ja sowieso keinen Spa und ein eigener Telefonanschlu war fr mein Studentenwohnheim erst noch in Planung.

1.2 Jetzt geht's los

Ein Jahr spter war es soweit, ein zweiter Versuch trat in Planung. Die Zeit war gekommen, zum Schraubendreher zu greifen und eine zweite Festplatte zu implantieren. Mittlerweile hatten die Festplatten die Gigabyte-Grenze hinter sich gelassen. Hatte ich fr 40 Megabyte zu ATARI-Zeiten einmal umgerechnet 750 Euro bezahlt, kosteten mich diesmal fast vier Gbyte nur ein Drittel davon. Hier war nun endlich Platz genug, um Linux nicht nur auszuprobieren, sondern als Komplettsystem mindestens gleichberechtigt neben Windows zu installieren. Mittlerweile war die Kernelversion auch um eine Nummer weiter gehpft und von den Installationsprogrammen hrte ich nur Gutes. Da traf es sich hervorragend, das S.u.S.E.⁴ ihr Linux in der Version 5.1 angekndigt hatte. Das wurde in verschiedenen Fachzeitschriften gerade fr blutige Laien sehr empfohlen⁵, also bestellte

²Die dreiteilige Versionsnummer kennzeichnet einen Kernel einwandfrei und gibt darber hinaus noch Hinweise fr den gebten Betrachter.

³natrlich heit es X-Window, ohne 's', aber als Anfnger wute ich das noch nicht

⁴Damals hie SuSE noch S.u.S.E.. Die Punkte kamen spter abhanden, was von einigen SuSEianern immer noch bedauert wird. Nheres dazu auf <http://www.goodbye-dots.de>.

⁵Erinnert sich noch jemand an Yggdrasil oder DLD? Das waren zwei weitere Linux-Distributionen dieser Zeit, die aber eben nicht mit soviel gedrucktem Papier ausgeliefert wurden. S.u.S.E. bot damals neben der eigenen Distribution noch die Mglichkeit, sich aus einem auf CD's gebrannten ftp-Server-Abbild sein Linux selbst zusammenzustellen. Diese Vielfalt fand ich als Linux-Anfnger ziemlich verwirrend

ich es, standesgemäß per eMail. Die zwei Wochen Lieferzeit waren dann aber sehr lang, ich muß sowas eigentlich immer sofort haben, wenn ich mich zu solch einem Schritt entscheide.

Pünktlich zu meinem Geburtstag war es soweit, der Paketbote brachte neben einem Geburtstagskuchen von meiner Mutter auch S.u.S.E.'s selbstgebranntes Linux, vier CD's und ein großes schwarzes Handbuch, Kernelversion 2.0.32. Einen Abend später machte ich mich dann ans Studium des Handbuchs und den ersten Installationsversuch. Mehr als ein Versuch wurde es allerdings auch nicht. Nach drei Stunden hatte ich mich, Gott, die Welt im allgemeinen und S.u.S.E. im speziellen verflucht. Frustriert verließ ich den Schauplatz meiner Niederlage. Es war mir zwar gelungen, mittels Bootdiskette ins Hauptmenü des Installationsprogramms zu kommen (eine sehr frühe Version von YaST), viel weiter war ich allerdings noch nicht. Jedesmal, wenn die Anleitung selbstbewußt vom Neustart inklusive anschließendem automatischen Formatieren der ausgewählten Partitionen sprach, sah ich eine rote Warnmeldung, daß die Installation mit einem nicht näher spezifizierten Fehler abgebrochen worden war.

Also machte ich am nächsten Nachmittag mit meiner besten Freundin (genauso computerverrückt wie ich) einen weiteren Versuch. Wiederum drei Stunden später, nachdem wir alles erdenkliche einschließlich Hardwareänderungen ausprobiert hatten, brachen wir auch diesen Versuch deprimiert ab und sie fuhr etwas desillusioniert nach Hause. Dabei wollte sie doch auch noch Linux installieren. Aber wenn wir das nicht mal mit vereinten Kräften schafften, wen hätten wir noch fragen sollen? Die einzigen Bekannten mit *Computererfahrung* waren wir selbst.

Wir hatten uns dem Problem von allen Seiten genähert und waren auch eine Fehlermeldung weiter, aber Linux weigerte sich weiterhin standhaft, die nötigen Partitionen einzurichten und sich zu installieren. Dabei hatte ich eine komplette Festplatte freigeräumt und sogar schon unter DOS in die richtigen Partitionen aufgeteilt. Als Sabine gegangen war, startete ich einen letzten Versuch, diesmal wieder von Diskette und auf die neue, erste Festplatte. Mein System wies nämlich die Eigenart auf, daß diese erste Festplatte als Single auf dem primären IDE-Port hockte, während sich die zweite Platte und das CD-ROM-Laufwerk den sekundären Port teilten. Damit wollte ich das Dilemma mit Schreib-/Lesefehlern vermeiden, das mich schon einmal eine Western Digital Platte gekostet hatte. Die entstehende Lücke, eben die fehlende `hdb`, schien Linux doch sehr zu irritieren, wenn es dahinter installiert werden sollte. Na jedenfalls, die wackere Helferin war noch nicht ganz aus der Tür, da lief die Installation. Diese Hürde war also genommen. Hätte ich eine Liste aufstellen sollen, wieviel Hürden noch zu nehmen waren, ich wäre sicher eine kleine Weile beschäftigt gewesen.

Auf meiner To Do Liste ganz oben stand das Einrichten des X-Servers. Dabei habe ich dann mein schönes neues Linux-System auch gleich richtig abgeschlossen. Schuld war meine Maus (die vom Computer natürlich). Die hatte ich just zum Geburtstag bekommen, eine Logitech Mouseman+ mit einem sinnreichen

Rädchen als mittlerer Maustaste (sie tut auch jetzt, nach mehr als vier Jahren noch ihren Dienst). Die Maus funktionierte unter WIN95 prächtig (Ehrenwort), nur für Linux war sie wohl etwas zu neu. Das Logitech-Protokoll passte nicht, Mouseman und Microsoft wollten auch nicht mit `/dev/psaux` laufen. Und an ein funktionierendes Scrolling per Mause rad wagte ich überhaupt nicht zu denken.

Bei einer dieser Proberaktionen ist es dann passiert, ich kam mit dem Ellenbogen an die Maus und es war aus. Keine Taste half mehr, das System hatte sich erhängt. Als nach einer Viertelstunde immer noch nichts passiert war, blieb nur noch der Griff nach der Reset-Taste. Nun folgte Überraschung Nummer zwei, ich hatte scheinbar auch `lilo`⁶ nicht richtig konfiguriert. Mit einem einsamen L blieb der Rechner beim Booten stehen. Nachdem ich wieder ein Viertelstündchen hoffnungsvoll gewartet hatte, war es Zeit, die DOS-Bootdiskette rauszukramen, um wenigstens den winzigweichen Teil meiner Daten zu retten. `FDISK /mbr` rettete mir den Zugang zu meinem Rechner, aber wo war jetzt wohl mein Linux? Na ja, bei den CD's lag ja auch eine Bootdiskette und im Startmenü gab's die Möglichkeit, ein schon installiertes System zu starten, vorausgesetzt, man weiß die Bezeichnung der root-Partition.

Beim dritten Versuch stimmte dann auch das und ich war nach ca. einer Stunde wieder im Konfigurationsmenü für den X-Server. Diesmal hatte ich die Maus weit, sehr weit weg gelegt und probierte es mit PS/2-Maus an `/dev/psaux`. Das klappte einwandfrei, die Grafikkarte war auch schnell eingestellt und schwups, der X-Server lief und ich war in `fvwm2`. Das führte dann gleich zum nächsten Punkt auf meiner Wunschliste, `TeX` installieren, bzw. in all dem Linux-Gewusel wiederfinden. Aber, oh Wunder, alles schon da, mit X-Shell und Editor. Und sogar schon fertig verdrahtet, d.h. laden, losschreiben und wohlfühlen.

Also zum nächsten Punkt auf meiner Liste, den Emulatoren. Aufmerksames Studium des Lieferumfangs der S.u.S.E.-Distribution hatte mich auf `STonX` gebracht, einen ATARI ST Emulator für Linux. Im Familienstammsitz tat zu dieser Zeit immer noch mein Diplom-ATARI Dienst, d.h. ab und zu durfe ich noch Systempflege betreiben und die alten Spiele spielen. Das meiste gibt's ja mittlerweile als Portierungen auch für andere Systeme, nur `Ballerburg`, hatte ich auf dem PC schmerzlich vermißt. Es war das erste Spiel, das ich je auf meinem ersten eigenen Computer gespielt habe, da habe ich wohl eine sentimentale Bindung entwickelt. Also schnell nochmal in den DOS-Modus gewechselt und per Modem ein TOS-Image besorgt. Nun hatte ich also den Emulator und TOS 2.06 (genau wie auf meinem Mega ST). Was ich noch nicht wusste, ich hatte auch schon `Ballerburg`! Es war bei `STonX` dabei, mein Glück war kaum zu fassen. Also war das erste, was ich auf meinem Linux-P133-Rechner spielte, ein altes 8bit-Spiel aus den späten 80er Jahren. Es war einfach wunderbar. Damit war ich in meiner imaginären Liste wieder einen Punkt weiter und jetzt wurde es etwas schwieriger.

⁶LinuxLOader, ein kleines Progrämmchen, das sich z.B. im Master Boot Record der Startpartition einnisten kann, um wahlweise Linux oder Windows zu booten

Soundkartenunterstützung, PPP, Netscape und der DOS-Emulator standen noch aus. Netscape zu installieren war kein Problem, nachdem ich herausbekommen hatte, wie man unter Linux tgz-Archive entpackt. Dafür hatte ich mir ja nun extra S.u.S.E. Linux angeschafft, schließlich wurde das mit dem dicksten und laut Fachpresse besten Handbuch geliefert. PPP und Soundkartenunterstützung strich ich nach meinem ersten gescheiterten Kernel-Kompilierungsversuch erstmal wieder von der Liste bzw. verschob es auf später. Zu 5.1-Zeiten war die Modularisierung des Kernels noch nicht so weit. Für jeden Soundkartentyp musste ein extra Kernel her und um ein Modem zu betreiben, mußte der PPP-Support auch erstmal in den Kernel kompiliert werden. Damit qualifizierte sich Linux zumindest damals noch als echtes Abenteuerer- bzw. Experten-System, jedenfalls in meiner damaligen Einschätzung.

Aber es gab ja auch so schon genug zu entdecken, schließlich hatte ich ein völlig neues Betriebssystem mit perfekt funktionierendem T_EX, einer neuen, ungewohnten GUI⁷ und einer ganzen Festplatte voll von Programmen, die auch erst mal erforscht werden wollten. Voller Elan machte ich mich daran, diesen Bericht zu verfassen. Natürlich in T_EX, denn mit MS Word hatte ich mich während meiner Dissertation genug herumgeschlagen. Es war fast wie nach Hause kommen, ich konnte vor mich hinT_EXen und zur Entspannung Ballerburg im STonX-Fenster spielen. Ach ja, und nebenbei noch etwas für die Disputation (Doktorprüfung) lernen...

1.3 Herkunft der Texte

Wer dieses Buch aufmerksam betrachtet, dem wird ein Teil der Kapitel bekannt vorkommen. Zwischen Juni 2000 und August 2001 entstand im Auftrag von SuSE ein Linux Knowledge Portal, dem viele der hier enthaltenen Kapitel entstammen. Seit September 2001 wird dieses Linux Knowledge Portal (<http://linuxkp.org>) von ein paar Freunden und mir auf privater Basis weitergeführt. Da sich der selbstgewählte Auftrag des Linux Knowledge Portal mit dem Entstehungszweck dieses Buches deckt, habe ich schamlos alle passenden Artikel überarbeitet und als Buchkapitel recycled.

1.4 Motivationsgründe

Nun aber los, genug von meinen ersten Erfahrungen, kommen wir zum eigentlichen Grund dieses Buches. Sie haben sich entschieden, zu Linux zu wechseln oder denken zumindest ernsthaft darüber nach (sonst hätten Sie nicht bis hier her gelesen). Höchstwahrscheinlich werden Sie Ihre Gründe, sich mit Linux zu beschäftigen, in der folgenden Liste wiederfinden:

⁷Graphical User Interface, Benutzeroberfläche

- Ich habe genug von Bluescreens und will die vielgerühmte Stabilität von Linux auskosten
- Alle erzählen, wie cool Linux ist und ich will endlich mitreden können
- Ich will endlich der Lizenzpolitik kommerzieller Softwarehersteller entgehen
- Ich nutze sowieso hauptsächlich GPL-Software
- Ich will mit Linux Serverdienste realisieren
- Ich habe genug von Windows-Viren
- ...

OK, einige dieser Gründe mögen dem einen oder anderen eher abwegig erscheinen, aber tatsächlich gibt es mindestens ebensoviele Gründe, Linux zu installieren, wie es Linux-Distributionen gibt.

Die Stärke von Linux liegt in allem, was mit Vernetzung und Internet zu tun hat, denn genau dafür wurde es entwickelt. Das von Windows komplett verschiedene Sicherheitskonzept ist ein weiterer sehr guter Grund, zu Linux zu wechseln bzw. es zumindest für Netzwerkaufgaben einzusetzen.

Kapitel 2

Linux ist anders

Autor: *Kurt Garloff*

Dieser Artikel gibt einen kurzen Überblick über das Konzept und die Philosophie von Linux. Er richtet sich in erster Linie an Benutzer, die mit Un*x noch nicht vertraut sind und stattdessen an die Konzepte von MS-DOS oder MS-Windows gewohnt sind.

2.1 Das Konzept: Multiuser und Multitasking

Unter MS-DOS kann immer nur ein Benutzer mit einem Programm arbeiten. Dieses Konzept ist bei Windows weitgehend beibehalten worden, wenngleich mehrere Programme in verschiedenen Fenstern gleichzeitig laufen können. Bei Un*x stand von vorneherein die Idee im Vordergrund, daß mehrere Benutzer gleichzeitig arbeiten und mehrere Programme gleichzeitig nutzen. Das parallele Arbeiten mehrerer Benutzer kann dabei z.B. über mehrere Terminals oder über Netzwerk erfolgen. Ein Grund, warum die Netzwerkfähigkeiten von Un*x-Systemen sehr ausgereift und leistungsfähig sind.

Außerdem stellt Linux zusätzlich mehrere virtuelle Konsolen zur Verfügung, die durch ((**<Strg>-<Alt>-<F1>** bis **-<F6>**) anzusprechen sind.

2.1.1 Benutzerverwaltung

Damit die Daten der Benutzer nicht durcheinander kommen, muß das System wissen, wer gerade arbeitet, d.h. die Identität des Benutzers muß dem System bekannt sein. Daher besitzen alle berechtigten Nutzer einen sogenannten Account und müssen sich einloggen. Dies geschieht in der Regel durch Eingabe des Benutzernamens und eines Passwortes. Das System weiß dann, daß dieser Benutzer (repräsentiert durch eine Zahl, die Benutzernummer, User-ID, UID) aktiv ist. Jeder Benutzer besitzt ein eigenes Verzeichnis, das sogenannte home-Verzeichnis,

in dem er seine Dateien ablegen kann. Außer der Benutzeridentität gehört der Benutzer noch einer oder mehrerer Gruppen an. Gruppen können benutzt werden, um Dateien zwischen mehreren Benutzern zu teilen. Geben Sie `id` ein, um Ihre Identität und Gruppenzugehörigkeit herauszufinden.

Es gibt einen Benutzer mit Sonderrechten: Den Systemverwalter. Nur er kann auf alle Dateien zugreifen, bestimmte Operationen durchführen. Er ist dafür zuständig, das System zu konfigurieren, die Programme zu installieren und Benutzer einzurichten. Dieser Verwalter heißt auf Un*x-Systemen traditionellerweise *root*. Er ist allmächtig: Er bestimmt, was andere Benutzer machen dürfen und was nicht. Während ein normaler nichtprivilegiertes Benutzer nicht ohne weiteres Schaden (außer für sich selbst) in einem vernünftig konfigurierten Un*x-System anrichten kann, trifft *root* eine große Verantwortung: Ein falscher Befehl kann zum Nichtfunktionieren wichtiger Systemfunktionen oder gar zum Verlust aller Daten führen. Daher sollte der Systemverwalter

- Un*x-Erfahrung haben bzw. erwerben
- zusätzlich zu seinem *root*-Account noch einen normalen Benutzeraccount haben, den er normalerweise benutzt, und sich nur als *root* einloggen, wenn es erforderlich ist
- darauf achten, daß kein anderer (potentiell unerfahrener oder bössartiger) Benutzer sich Zugang als *root* verschaffen kann
- sich darüber bewußt sein, daß im Internet Hacker lauern, die darauf warten, sich zu angeschlossenen Computern unberechtigterweise Zugriff zu verschaffen.

Um zu vermeiden, daß verschiedene gleichzeitig laufende Programme oder verschiedene Benutzer einander beeinträchtigen können, implementieren Un*x-Systeme ein *Schutzkonzept*. Ein wichtiger Aspekt dieses Schutzkonzeptes ist, daß jedes Programm einen eigenen Speicherbereich besitzt und nur auf diesen zugreifen kann und daher andere Programme auch nicht in unkontrollierter Weise beeinflussen kann. Moderne Prozessoren (bei ix86 ab 286) besitzen Mechanismen, um diesen Schutz zu implementieren. Dafür gibt es privilegierte Befehle, die nur in einem *privilegierten Prozessormodus* ausgeführt werden können. Entsprechend ist das Betriebssystem zweigegliedert: Es gibt den *Kernel* (Betriebssystemkern), das Betriebssystem im engeren Sinne, *Linux*, welches den privilegierten Prozessormodus nutzt, und Programme, welche im nichtprivilegierten Prozessormodus ausgeführt werden. Da nur der Kernel die privilegierten Befehle nutzen kann, kontrolliert er alle Programme.

Einerseits ist der Kernel für die Verwaltung der Ressourcen (Hardware) zuständig, d.h. er verwaltet und verteilt Arbeitsspeicher, Prozessorzeit, Festplattenzugriffe, Zugriffe auf Soundkarten usw. auf die verschiedenen Aufgaben. Diese

Aufgaben sind einerseits die Prozesse, andererseits auch Dinge, die der Kernel selbst erledigt, z.B. das Cachen (Zwischenspeichern) von Festplattenzugriffen im Arbeitsspeicher. Andererseits stellt er den Programmen Dienste zur Verfügung, die diese per *Systemaufruf* nutzen können. Vom Kernel her kommt auch die Unterscheidung zwischen normalen Benutzern und root.

Genau genommen bezeichnet Linux nur den Kernel, während die meisten anderen Programme und Bibliotheken für viele Un*x-Varianten existieren und zu einem großen Teil aus dem *GNU-Projekt* der Free Software Foundation (FSF, <http://www.gnu.org>) stammen. Erst die Kombination aus Kernel und einer Sammlung von Programmen, Bibliotheken und dem Compiler macht aber ein nützliches System aus.

2.2 Programmphilosophie

Un*x existierte lange bevor graphische Benutzeroberflächen in Mode kamen. Aus diesem Grund kommt dem Befehlsinterpreter, der *Shell* (unter DOS die Eingabeaufforderung), bei Un*x eine sehr hohe Bedeutung zu. Entsprechend dieser Bedeutung sind die unter Un*x verbreiteten Shells sehr leistungsfähig. Dies hat dazu geführt, daß sich eine Programmstruktur herausgebildet hat, die von der bei Windows bekannten sehr unterschiedlich ist. Während bei letzterem der Trend dahin geht, immer mehr Funktionen in ein Programmpaket zu integrieren (MS-Office, Betriebssystem mit integriertem Browser), wurde unter Un*x die Philosophie kleiner, spezialisierter und flexibel einsetzbarer Programme verfolgt.

Die Trennung in kleinere Einheiten hat einen entscheidenden Vorteil: Die Funktionalität und die Größe des Codes der Einzelkomponenten ist so klein, daß die Hoffnung, diese weitestgehend von Fehlern zu befreien, nicht von vornherein aussichtslos ist. Die Anzahl der nützlichen Tools ist sehr hoch und es ist schwierig sie alle zu kennen. So entdeckt auch ein erfahrener Systemadministrator manchmal neue Hilfsprogramme, die schon lange existieren. Die Schlüsselfrage unter Linux ist oft: Wie heißt das Programm (oder die Programme), mit denen ich diese Aufgabe erledigen kann und wo find ich es? Hilfsprogramme wie `locate`, `find`, `whatis` und `apropos` helfen einem aber, wenn man eine Idee hat, wie der gewünschte Befehl heißen könnte oder welche Stichworte ihn charakterisieren. Meistens gibt es mehrere Möglichkeiten, ein Problem zu lösen.

2.2.1 Dateisysteme und Dateisystemstruktur

Die verschiedenen Datenträger und Partitionen leben bei Un*x in nur einem Namensraum: Es gibt keine Laufwerksbuchstaben, das oberste Verzeichnis heißt nicht C oder D, sondern / (slash), auch dies eine große Umstellung im Vergleich zu DOS/Windows. Stattdessen werden die Dateisysteme auf den Datenträgern beim Booten durch den Befehl `mount` im Dateisystembaum befestigt (gemoun-

tet). So bewirkt `mount /dev/sr0 /cdrom`, daß von nun an alle Dateien im Verzeichnis `/cdrom/` und Unterverzeichnissen von dem Gerät `/dev/sr0` (erstes SCSI CD-ROM Laufwerk) gelesen werden (im Verzeichnis `/dev/` stehen immer die verschiedenen Geräte). Wie man sieht, werden Verzeichnisse stets mit dem Slash `/` abgetrennt, nicht mit dem Backslash. Ein `umount` hebt eine solche Zuordnung wieder auf, hängt also ein Gerät oder ein anderes Teildateisystem wieder aus.

Für fest eingebaute Datenträger ist das Mounten sehr vorteilhaft. So kann z.B. der Inhalt einer Festplattenpartition auf eine andere verschoben werden. Danach wird die neue Festplattenpartition gemountet und schon ist alles wie vorher, bis auf einen geänderten Eintrag in der Tabelle `/etc/fstab`, wo eingetragen ist, welches System wo gemountet ist. Für entfernbare Datenträger wie Disketten oder CD-ROMs ist das ein wenig unbequem. Anstatt einfach darauf zugreifen zu können, muß man nun jedesmal mounten und vor dem Entfernen wieder umounten. Es gibt aber gewisse Tools, die diese Unbequemlichkeit beseitigen:

- `mtools` - Erlauben den Zugriff auf Disketten mit MS-DOS (bzw. Win95) Dateisystem, ohne mounten zu müssen.
- `autofs` - Mountet und umountet Dateisysteme automatisch, wenn Datenträger vorhanden sind und auf das vorgesehene Verzeichnis zugegriffen wird.

Im Dateisystem von Un*x werden auch die Zugriffsrechte der Dateien gespeichert. Jede Datei gehört einem Benutzer und einer Gruppe. Es ist festgelegt, wer alles auf die Datei in welcher Art zugreifen darf. Dabei wird zwischen dem Eigentümer (`u=user`), der Gruppe (`g=group`) und den anderen (`o=others`) unterschieden und zu jeder dieser drei Möglichkeiten gibt es das Leserecht (`r=read`), das Schreibrecht (`w=write`) und das Ausführungsrecht (`x=execute`). Schauen Sie sich mal mit `ls -l` ein Verzeichnis an!

Die Struktur des Dateisystems sieht bei allen Linux-Installationen ziemlich ähnlich aus. Während z.B. unter `/home` alle Benutzerverzeichnisse liegen, stehen unter `/etc` die Systemeinstellungen, die ausführbaren Programme befinden sich in `/bin`, `/usr/bin` und `/usr/X11/bin`, Bibliotheken in `/lib`, `/usr/lib` und `/usr/X11/lib` usw. Dateien, die mit einem Punkt `.` anfangen, werden von `ls` standardmäßig nicht angezeigt (versteckte Dateien). Die Option `-a` ändert das.

2.2.2 Philosophie freier Software

In der Un*x-Welt gab es traditionell viele unterschiedliche Plattformen, die natürlich nicht binärkompatibel sein konnten. Daher war die natürlichste Art, Software zu verteilen, den Quellcode auszuliefern. Mit dem Vordringen kommerzieller Interessen in die Un*x-Welt, wurde mit dieser Tradition jedoch gebrochen und jeder Hersteller verkaufte seine eigenen Codes. In der akademischen Welt lebte die Tradition aber weiter.

Die Open-Source Bewegung ging und geht sogar noch einen Schritt weiter: Sie hebt die Bedeutung von offenem und kostenfreiem Quellcode für einen erfolgreichen Entwicklungsprozeß ausdrücklich hervor. Einige Vorteile sind:

- Es gibt wesentlich mehr Entwickler und Tester als bei einem geschlossenen Projekt.
- Das Finden und Korrigieren von Fehlern geht schneller und effizienter.
- Die Rückkopplung zwischen Entwicklern und Anwendern klappt besser.
- Designentscheidungen stehen zur Debatte und oftmals werden schlechte Entscheidungen rechtzeitig bemerkt und korrigiert.
- Code aus anderen Projekten kann benutzt werden, so daß das Rad nicht immer wieder neu erfunden werden muß.

Außer dem offenen Quellcode und der Kostenfreiheit ist mit Freier Software noch etwas mehr gemeint: Die Freiheit, mit der Software tun und lassen zu können, was man möchte. Diese Freiheit ist bei vielen Programmen durch die GNU General Public License (GPL) geschützt. Man darf die Software nach Belieben verändern, für seine eigenen Zwecke einsetzen, sie weitergeben usw. Ein paar Einschränkungen gibt es aber, um diese Freiheit zu schützen:

- Von GPL-Code abgeleitete Software muß wieder unter der GPL stehen.
- Der Quellcode muß zugänglich gemacht werden, und zwar umsonst.
- Haftungsausschluß: Die Autoren freier Software lassen sich für die Arbeit, die sie meistens ohne finanzielle Gegenleistung erbracht haben, nicht auch noch verklagen, wenn irgendetwas schief läuft.

Die Gründe, warum Programmierer Software programmieren, ohne Geld dafür zu erhalten, sind vielfältig und bei jedem Programmierer verschieden. Einige Gedanken sollen hier aufgezählt werden:

- Der Programmierer erhält dafür von anderen, die auch freie Software entwickeln, gute Programme, und muß diese nicht selbst schreiben.
- Die eingesetzten Tools sind auch frei, daher braucht er weniger Einkünfte. Die einzig nennenswerte *Investition*, die zur Softwareentwicklung benötigt wird, ist die Arbeitszeit.
- Durch Mitarbeit an einem Softwareprojekt kann man sich einen guten Ruf erarbeiten. Die Freie-Software-Szene wird oft als *Community* bezeichnet.

- In der Weltanschauung des Programmierers ist eine Welt, in der die Gesellschaft das Recht an einer Schlüsseltechnologie, der Softwaretechnologie, besitzt, wesentlich wünschenswerter als eine, in der diese von einer oder wenigen Firmen kontrolliert wird.

Trotzdem müssen auch Programmierer leben. Manche haben einen Job in der Softwarebranche, wo sie nach dem klassischen Modell proprietäre Software programmieren. Andere sind hauptberuflich Akademiker oder Studenten. Weitere leben von Support und Consulting.

Zunehmend gibt es aber auch Firmen, die bereit sind, für die Entwicklungskosten von Software aufzukommen, obwohl die Software anschließend als Freie Software für alle nutzbar ist. Aber was soll die Firma tun, wenn sie diese Software braucht und kompetente Entwickler nur Freie Software herstellen wollen? Dafür sind die Entwicklungskosten geringer (keine kostenpflichtigen Entwicklungswerkzeuge müssen bezahlt werden), das Ergebnis aufgrund des Open-Source Modells besser und die Pflege problemloser.

Kapitel 3

Leitfaden für Umsteigewillige

Autor: *Frank Rennemann*

Allein die Tatsache, daß Du diesen Artikel liest, lieber Leser, beweist schon, daß Du zumindest ein gewisses Interesse an Linux mitbringst, wenn Du nicht schon längst mit Linux arbeitest. Selbst im ungünstigsten Fall ist die erste Bedingung für einen Wechsel zu Linux schon erfüllt: das Interesse! Auf die anderen Bedingungen bzw. den Umstieg erleichternden Umstände wird im Folgenden eingegangen.

3.1 Die Hardware

Was braucht man außer Interesse noch? Nun ja, ein Computer wäre hilfreich, schließlich will man das beste aller Betriebssysteme ja auch irgendwo unterbringen. Und wenn schon ein Computer, wollen wir uns hier auf das meistverbreitete Modell beschränken, den IBM-kompatiblen PC. Was das für ein PC sein soll? Eigentlich ziemlich egal, die Spannweite reicht von alten Pentium-Rechnern bis zu ultramodernen Pentium 4 bzw. Athlon XP Maschinen. Die wichtigsten Eckdaten unterscheiden sich je nach Einsatzzweck allerdings z.T. erheblich.

Auf einem älteren Pentium-Rechner (66-233 MHz, Baujahr 95-99) mit 64 MB RAM und mindestens einem GB Festplattenplatz kann man schon erste Schritte machen. Allerdings sollte man sich dann etwas Zeit mitnehmen, wenn man eine graphische Oberfläche wie KDE oder Gnome ausprobieren will. Es gibt ein paar andere Windowmanager, die sehr gut und einigermaßen performant auf diesen alten Rechnern laufen, z.B. WindowMaker oder Blackbox, aber die meisten Wechselwilligen möchten ja eigentlich genauso weiterarbeiten, wie sie das von Windows gewohnt sind.

In diesem Fall wird hoffentlich auch ein einigermaßen aktueller Rechner vorhanden sein, ein Pentium 2 (333-500 MHz, Baujahr 98-2000) PC mit mindestens 64, besser 96-256 MB RAM und 4-10 GB freiem Festplattenplatz (Standardrechner vor ca. drei Jahren) ergibt schon eine gute Linux-Workstation, auch für ak-

tuelle Linux-Distributionen. Bei taufischen Rechnern mit aktuellen Prozessoren und ultramodernen Grafikkarten sollte man sich vorher kundig machen, ob die Grafikkarte schon von Linux unterstützt wird. Gehen wir also mal davon aus, das ein passender PC zur Hand ist. Was fehlt noch?

3.2 Die Software

Ganz klar, eine Linux-Distribution. Wo nimmt man die her? Nun, anders als bei rein kommerziellen Betriebssystemen wie **Microsoft Windows** oder **MacOS** gibt es bei **Linux** viele verschiedene Möglichkeiten, an Installationsmedien zu kommen. Da der Haupt-Entwicklungsweg von **Linux** das Internet war und ist, liegt hier natürlich auch die schnellste Möglichkeit, an sogenannte ISO-Images zu kommen, die man sich dann selbst auf CD brennen kann. Diese ISO-Images gibt es außer von **SuSE** von fast jeder aktuellen Distribution, wobei die kommerziellen Distributoren meist eine etwas abgespeckte Variante zum Download anbieten. Der Download solcher CD-Images empfiehlt sich allerdings nur, wenn man auch über DSL und am besten auch noch über eine Flatrate verfügt. So ein Download braucht pro CD-Image ca. zwei Stunden und eine Distribution wie **Redhat** oder **Mandrake** hat jeweils ca. drei CD-Images. **SuSE** geht einen Sonderweg, indem es seine Distributionen nicht als ISO-Images zum freien Download anbietet, sondern stattdessen vier bis acht Wochen nach Erscheinen einer neuen Distribution diese als ftp-Version direkt zum Installieren anbietet.

Hat man keine DSL-Flatrate, kann man sein Glück über den Umweg eines ISO-Image-Versenders (z.B. <http://www.linux-iso.de/linux/index.html>) versuchen. Beiden Wegen ist gemeinsam, daß man an alle frei verfügbaren Distributionen herankommt, meist sehr aktuell und das man nicht durch Handbücher in gedruckter Form belästigt wird. Wer auf Gebundenes und CD's in geschmackvollem Design oder auch DVD's nicht verzichten will, sollte einen Besuch beim nächsten Buchhändler oder EDV-Laden seines Vertrauens in Erwägung ziehen. Vereinzelt sollen Linux-Distributionen sogar schon in Tankstellen zum Verkauf angeboten worden sein. Nicht zuletzt kann man **Linux** natürlich auch über's Internet auch bei den bekannten Linux-Distributoren bestellen.

Den vielleicht besten Weg habe ich mir bewußt bis zum Schluß der Aufzählung aufgehoben. Wenn man Freunde oder gute Bekannte hat, die den Weg zum Licht schon gegangen sind, sind die bestimmt auch im Besitz einer oder mehrerer Linux-Distributionen. Eigentlich jeder Linux-Treibende, den ich kenne, hat schon mindestens ein Care-Paket an einen Freund weitergegeben, einfach mal so zum Ausprobieren. Da ist dann meist nicht nur die gedruckte Dokumentation dabei, sondern in den allermeisten Fällen kann man sogar so unverschämt sein, den Bekannten und sein Wissen zum eigenen Vorteil anzuzapfen. Wer ein Linux verleiht oder verschenkt, fühlt sich meist auch für die gelungene Erstinstallation verantwortlich. Wir Linux-Treibende sind nämlich ungemein gesellig und lieben

es, für einen Kaffee oder Bier und Pizza eine Erstinstallation fachmännisch zu betreuen...

3.3 Welches Linux soll ich nehmen?

Ich sehe schon Deine nächste Frage voraus, lieber Leser, welche der zahlreichen Distributionen soll ich denn nehmen? Welche ist die Beste? Mandrake? Debian? Redhat? SuSE? Oder eine von den zahlreichen anderen? Das ist eine Frage, über die sich Kenner, Halbkenner und Laien tagelang ohne Ermüdungserscheinungen in die Haare geraten können, ohne einer Antwort auch nur im mindesten näher zu kommen. Auch wenn die Frage auf den ersten Blick schwierig zu beantworten scheint, die Antwort ist eigentlich ganz einfach. Es ist mehr oder weniger egal, man wird mit jeder von Ihnen ein lauffähiges System hinkriegen und jede Linux-Installation ist besser als Windows.

Im Ernst, wer es einfach haben will, nimmt Mandrake, Redhat oder SuSE, wer hohe Ansprüche an sich selbst hat und das System von Anfang an gründlich kennenlernen will, kann sich auch an Debian versuchen. Die ganz Harten nehmen Gentoo oder Linux from Scratch und bauen sich ihr System komplett selbst. Einsteiger sollten sich diesen ziemlich dornigen Weg allerdings ersparen. Wem es auf gute Handbücher ankommt, der kommt eigentlich an SuSE nicht vorbei, deren Handbücher haben allgemein und zu Recht einen guten Ruf. Ein nicht ganz unwichtiges Kriterium der Entscheidung für eine Distribution sollte immer sein, ob man Bekannte hat, die mit einer bestimmten Distribution schon Erfahrung haben. Die würde sich dann auch für einen selbst zum Einstieg empfehlen, gemeinsam leidet es sich einfach leichter und bei jedem Einstieg gibt es Stolperfallen, um die einen jemand mit Erfahrung herumgeleiten kann.

3.4 Das planvolle Vorgehen

3.4.1 Eingebaute Komponenten

So, also, wir haben drei Dinge beisammen, Interesse, PC und Distribution, was fehlt noch? Ein paar Gedanken zum weiteren Vorgehen, denn ein ungefährer Schlachtplan hat noch nie geschadet. Es gibt noch ein paar Dinge zu bedenken, bevor man ans Werk geht. Zunächst wäre da an die in den Rechner eingebauten Komponenten zu denken. Bis auf sehr neue und sehr exotische Teile (z.B. brandneue 3D-Grafikkarten, High-End-Soundkarten) sollten alle Innereien von Linux unterstützt werden. Wer sich nicht sicher ist, kann einen Blick in die Hardware-Datenbank von SuSE werfen (<http://hardwaredb.suse.de/>), auch wenn deren Aktualität in den letzten Jahren etwas abgenommen hat.

3.4.2 Notebooks

Bei Notebooks wird es sehr wahrscheinlich Probleme mit eingebauten Modems geben, meist handelt es sich um Win-Modems, von denen erst ein kleiner Teil durch Linux unterstützt wird (unter <http://www.linux-laptop.net/> gibt es diverse Installationsberichte und Tipps für Linux auf Notebooks). Auch das Powermanagement und Funktionen wie Suspend to RAM werden wahrscheinlich nicht funktionieren. Die Linux-Unterstützung der Hersteller ist bei Notebooks noch nicht sehr ausgeprägt. Gerade bei APM und ACPI, den zwei verschiedenen Standards zum Power Management, hakt es teilweise noch mächtig, weil sich die Hardwarehersteller oft nicht an die selbstdefinierten Standards halten. Erwähnenswerte ACPI-Unterstützung wird es wahrscheinlich erst mit dem Kernel 2.6 geben, den Linus für Sommer 2003 angekündigt hat. Schwierigkeiten können auch Soundchips und Grafikkarten machen. In einigen seltenen Fällen muß man dann auf 3D-Beschleunigung oder gar auf Sound verzichten.

3.4.3 Peripheriegeräte

Wie sieht's mit der Peripherie aus? Drucker? Sollten eigentlich alle funktionieren, mir ist noch keiner begegnet, der die Druckausgabe unter Linux verweigert hätte. Selbst GDI-Drucker, die für Windows entwickelt wurden, laufen meist unter Linux. Sogar einige der All-in-One Lösungen mit FAX-Modem, Drucker und Scanner sind unter Linux lauffähig.

Wie steht's mit den Scannern? Solange es sich um ein SCSI-Gerät handelt, ist man im grünen Bereich, die werden zum überwiegenden Teil unterstützt. Bei Parallelport- und USB-Scannern sieht es etwas anders aus, was natürlich wieder daran liegt, daß einige Hersteller es irgendwie nicht für nötig halten, mehr als Windows zu unterstützen. Canon ist so ein Beispiel. Einen Eindruck über unterstützte Scanner gibt <http://www.mostang.com/sane/sane-backends.html>.

Weiter im Text, wie sieht es mit externen ISDN-Modems aus? Finster, vor allem, wenn es sich um Telekom-Geräte handelt, da die Telekom kein offenes Protokoll für ihre ISDN-Modems verwendet, allerdings auch keine Informationen dazu an freie Entwickler herausgibt. Im grünen Bereich ist man mit ISDN-Karten (z.B. AVM FritzCard), oder Netzwerkkarten aller Art für DSL. Auch alle handelsüblichen normalen Modems werden sich ansprechen lassen.

3.5 ...und die alten Daten?

3.5.1 Backup, Backup und nochmal Backup

Obwohl es unwahrscheinlich ist, daß es zu Datenverlusten kommt, wenn man sich an die Installationsanleitung des zu installierenden Linux hält, ist es natürlich trotzdem oberstes Gebot, vor der Installation eines neuen Betriebssystems alle

Daten, an denen man hängt, irgendwo zu sichern, wo ihnen nichts passieren kann. Die Auswahl der Daten ist dabei nicht schwer, alles, was man selbst erstellt hat, wird man nicht über irgendwelche Installations-CD's wiederherstellen können. Diese Daten gehören also zuerst gesichert. Steht genügend Platz und Zeit für ein Vollbackup zur Verfügung, sollte man sich die Zeit dafür auch nehmen.

Der Rechner, auf dem später Linux installiert werden soll, ist dabei in jedem Fall die falsche Wahl für die zu sichernden Daten. Ein Backup auf CD oder einem anderen Rechner ist kein Luxus, auf den man verzichten kann, sondern eine Notwendigkeit, auf die man im besten Fall nie zugreifen muß, die man aber unbedingt haben sollte.

Bei der Gelegenheit sollte man sich auch gleich Gedanken machen, ob man in Zukunft diese Backups nicht besser in regelmäßigen Abständen machen sollte. Ein Privathaushalt ist zwar keine Firma, wo bei Datenverlust gleich Verdienstausschlag droht, trotzdem ist es ärgerlich, wenn durch einen Festplattencrash oder gar den Diebstahl des Rechners die Urlaubsbilder der letzten paar Jahre, alte Steuererklärungen und Briefe oder eben auch nur mühsam errungene Spielstände verloren sind.

3.5.2 Die anderen Mitbewohner der Boot-WeeGee

Was passiert mit einem etwa schon vorhandenen Windows, das auch weiter existieren soll? Es kann bleiben, wo es ist, vorausgesetzt, auf der Festplatte ist noch genug Platz für die Linux-Partitionen oder es existiert sogar eine zweite Festplatte für Linux. Dank grub und lilo (zwei Bootprogramme für Linux) kann man dann schon beim Starten des Rechners entscheiden, welches Betriebssystem geladen werden soll.

Und die Daten aus der Windows-Installation, wie kann man darauf zugreifen? Solange die Windows-Partitionen FAT16 (DOS, Windows 95) oder FAT32 (Windows 95/98/98SE/ME/XP) als Dateisystem benutzen, hat man auch unter Linux vollen Zugriff auf diese Partitionen. Meist werden entsprechende Verknüpfungen schon bei der Linux-Installation angelegt, so daß sie bei Bedarf sofort zur Verfügung stehen. Und die bekannten Distributionen bringen Tools mit, mit denen vor der Linux-Installation die FAT-Partitionen verkleinert werden können, um Platz für Linux zu schaffen. Problematisch wird es bei NTFS-Partitionen, wie sie von WindowsNT4, 2000 Pro und XP angelegt werden können. Es gibt die Möglichkeit des lesenden Zugriffs, die derzeitigen Lösungen zum schreiben Zugriff von Linux aus auf NTFS-Partitionen ist bestenfalls experimentell zu nennen. Das liegt auch hier nicht an Linux bzw. dessen Entwicklern. Eine gewisse Firma, die mit winzigen Weichlingen handelt, läßt sich bei NTFS nur ungern in die Karten sehen. Wem etwas an seinen Daten liegt, der sollte auf derartige Experimente verzichten und lieber eine zusätzliche FAT32-formatierte Partition einrichten, die von beiden Betriebssystemen les- und schreibbar ist.

3.6 Man nehme...sich Zeit

Was fehlt nun noch? Etwas Zeit sollte man mitbringen, ein Betriebssystem installiert man nicht eben mal in der Kaffeepause zwischen zwei Keksen. Rein zeitlich ginge das schon, ein aktuelles SuSE Linux, in der Standardkonfiguration mit OpenOffice läßt sich von DVD auf einem einigermaßen modernen Rechner innerhalb von ca. einer halben Stunde installieren, wenn keine Probleme auftreten. So lange braucht oft schon eine Recovery-Installation von Windows und da hat man keinerlei Einflußmöglichkeiten.

3.6.1 Lesen, lesen, lesen...

Nein, die Zeit sollte man sich nehmen, um sich in den mitgelieferten Dokumentationen, egal ob gedruckt oder online, über sein neues Betriebssystem zu informieren. Vor allem über die einzelnen Installationsschritte informiert man sich am Besten vor dem ersten Einlegen der Installations-CD/DVD. Und auch ein wenig Lektüre über die ungewohnte neue Benutzeroberfläche kann nicht schaden. Denn natürlich kommt einem alles ein wenig verändert vor, auch wenn KDE und Gnome sich alle Mühe geben, einem den Übergang von Windows zu erleichtern. Noch etwas? Ein wenig Geduld sollte man aufbringen, wenn zu Anfang nicht alles so reibungslos geht, wie man sich das vorgestellt hat. Es ist wie in eine neue Stadt ziehen, man weiß, das es alles gibt, was es in der alten Stadt auch gab, aber man weiß noch nicht genau wo. Einfach mal die Hauptstraße (Menüleiste) langspazieren klärt schon viele Fragen.

3.7 In medias res

Bevor man diese Menüleiste zu Gesicht bekommt, ist nun allerdings noch eine letzte kleine Hürde zu nehmen, die Installation. Vorher noch ein letzter Blick in die Runde. Haben wir alles, was wir brauchen? Sind alle Hardwarekomponenten auf Lauffähigkeit mit Linux gecheckt? Gibt's einen bootfähigen Satz Linux-CD's? Liegt die alte Windows-Boot-Diskette bereit für den Fall, daß irgend etwas schiefgeht? Ist das frisch erstellte Daten-Backup an einer sicheren Stelle untergebracht und auch auf Funktionstüchtigkeit getestet (jedenfalls nicht auf dem PC, der zur Installation ansteht!!!)? Liegen die Handbücher bzw. Ausdrücke von der Installationsdokumentation griffbereit? Wo ist der gute Bekannte, der mit Rat und Tat zur Seite stehen kann? Steht eine Kanne Kaffee bzw. genug Bier/Cola/Wasser zur Verfügung? Wo ist die Telefonnummer des Pizzabringdienstes? Alles da? Dann mal los, die erste Installations-CD ins Laufwerk einlegen und der Aufbruch in eine neue Zeit kann beginnen...

3.8 Allgemeines zur Installation

Alle aktuellen Distributionen bringen mittlerweile Installationsroutinen mit, die dem Anwender das Partitionieren eines Linux-Systems abnehmen. Z.T. gehören auch Programme zum Ändern der Partitionierung eines bestehenden Windows-Systems dabei, mit denen bestehende Partitionen verkleinert werden können, um Platz für Linux zu schaffen.

Obwohl man also im besten Fall nicht mit der Entscheidung belästigt wird, wie der vorhandene Platz aufzuteilen ist, sollte man sich mit der Art und Anzahl der einzurichtenden Partitionen vertraut machen.

3.8.1 Partitionierung der Festplatte

Anders als z.B. Windows, wo der Swap-Speicher standardmäßig als Datei im Filesystem der Bootpartition liegt, wird für den Swap-Bereich bei Linux eine eigene Partition eingerichtet. Die Größe dieser Swap-Partition richtet sich in etwa nach der Größe des Arbeitsspeichers des Computers. Im Allgemeinen sind aber 128-256 MB genug für die Swap-Partition.

Daneben benötigt Linux noch eine kleine Partition für den Kernel, 5-10 MB sind meist genug, schließlich liegen hier nur der Kernel und seine Module. Als Mindestausstattung an Partitionen fehlt jetzt nur noch die eigentliche Arbeitpartition, der die meisten Installationsroutinen üblicherweise den gesamten restlichen freien Speicherplatz zuweisen.

Anders als bei Windows werden bei Linux keine Laufwerksbuchstaben verwendet, um Partitionen zu benennen. Das hat auf den ersten Blick schon zwei Vorteile. Erstens ist die Anzahl möglicher Partitionen dadurch nicht auf 26 (A-Z) beschränkt und zweitens kommen Verweise auf bestimmte Partitionen nicht durcheinander, wenn einzelne Partitionen aus dem System entfernt werden.

Unter Linux folgt die Benennung der Partitionen ihrer physikalischen Lage auf den Festplatten. Hierbei werden IDE-Festplattenpartitionen mit dem Kürzel `hd` benannt, Partitionen auf SCSI-Festplatten analog mit `sd`, beide gefolgt von einem Buchstaben, der die Reihenfolge der physikalischen Laufwerke im System angibt (üblicherweise `hda-hdd` bzw. `sda-sdn`. Dieser Benennung folgen außer den Festplatten auch alle anderen an den Controllern angeschlossenen Laufwerke (CD, DVD, ZIP etc.).

Die komplette Benennung einer Partition läßt sich daraus als Verzeichnispfad unterhalb von `/dev` als n-te Partition des X-ten Laufwerks darstellen, z.B. `/dev/hda1` für die erste Partition der ersten IDE-Festplatte. So, damit wäre die Hälfte schon geschafft, jetzt muß das Linux-System nur noch wissen, wo es diese Partition in den Verzeichnisbaum einhängen soll. Diesen Vorgang nennt man **mounten**. Die oben besprochenen Mindestpartitionen werden als `/boot` (`/dev/hda1`) sowie `/` (`/dev/hda3`) gemountet. Dem aufmerksamen Betrachter

wird nicht entgangen sein, daß `/dev/hda2` hier noch fehlt, die enthält die Swap-Partition und wird daher nicht in den Verzeichnisbaum eingehängt.

Die detaillierte Beschreibung dieser *Einhängpunkte* (Mountpoints) ist in der Datei `/etc/fstab` abgelegt. Hier steht dann auch für jede Partition, mit welchem Dateisystem sie formatiert ist. Auch hier hat im Vergleich zu **Windows** die wunderbare Vielfalt zugeschlagen. Gibt es dort nur drei Dateisysteme (FAT16, FAT32, NTFS), gibt es bei **Linux** deren gleich fünf, die sich mehr oder weniger größerer Verbreitung erfreuen, wobei sich darunter auch gleich vier sogenannte *journaling* Dateisysteme befinden.

Wer schon einmal **Windows** per Reset beenden mußte und die 20 Minuten danach auf den blauen Bildschirm gestarrt hat, während der Dateisystem-Check versucht hat, die aufgetretenen Inkonsistenzen im Dateisystem zu finden und zu beheben, wird den Einsatz eines *journaling* Dateisystems sehr zu schätzen wissen.

Da bei Journaling-Dateisystem kontinuierlich mitprotokolliert wird, wo gerade Änderungen vorgenommen werden, läuft der Dateisystem-Check im Ernstfall wesentlich schneller ab. Schließlich muß nur im Log nachgesehen werden, welche Dateien zur Zeit des Absturzes offen waren und gezielt repariert werden müssen.

Aktuelle Distributionen richten mittlerweile standardmäßig solche Dateisysteme ein, **Mandrake** und **Redhat** setzen `ext3` ein, **SuSE** hat schon früh auf **ReiserFS** gesetzt. Nach all dieser Theorie wollen wir uns einmal eine übliche Partitionierung eines Systems, auf dem außer **Linux** nichts existiert, ansehen. So oder ähnlich sieht

Tabelle 3.1: Partitionierung eines einfachen **Linux-only**-Systems

phys. Laufwerk	Mountpoint	Dateisystem
<code>/dev/hda1</code>	<code>/boot</code>	<code>ext2</code>
<code>/dev/hda2</code>	<code>swap</code>	<code>linux swap</code>
<code>/dev/hda3</code>	<code>/</code>	<code>reiserfs</code>

die Partitionierung aus, die nach der Installation der verbreiteten Distributionen auf der Festplatte ist. Plant man, verschiedene Distributionen auszuprobieren oder traut man den Beteuerungen nicht, das Updates problemlos möglich sind, ist man gut beraten, diese Standardpartitionierung etwas zu modifizieren. So ist es z.B. überhaupt nicht nötig und z.T. auch nicht sinnvoll, Programme einerseits und Daten andererseits in der selben Partition zu halten. Daten werden bei einem normalen **Linux**-System in einem gut einzugrenzenden Bereich des Dateibaums gespeichert, unter `/home`. Unterhalb dieses Verzeichnisses erhalten alle eingerichteten Benutzer einen nur für sie schreibbaren Bereich zugewiesen.

Damit bietet sich `/home` als Kandidat für eine eigene Partition an. Will man später evtl. das komplette System durch ein anderes **Linux** ersetzen, reicht es nach der Installation, die Partition mit den `/home`-Verzeichnissen wieder zu mounten und schon kann man mit dem neuen **Linux** auf die altgewohnten Daten zugreifen.

3.9 Linux zum Ausprobieren

Für all jene, die zwar schon einigermaßen versiert mit dem PC sind, ihr in langen Jahren gewachsenes System aber noch nicht mit einem neuen Betriebssystem erweitern wollen oder können, gibt es auch Linux zum sozusagen rückstandsfreien Ausprobieren. Neben der zu Unrecht oft und viel gescholtenen SuSE Live CD, die von SuSE als Ersatz der vorher verteilten Minimal-Distribution entwickelt wurde und seitdem auf Messen verteilt wird, gibt es als nichtkommerzielle Alternative z.B. Knoppix.

Knoppix ist eine Distribution, die auf Debian basiert und komplett von einer Boot-CD läuft. Einfach die Knoppix-CD ins CD-ROM-Laufwerk eines Rechners einlegen, neu booten und schon kann man mit Linux arbeiten, ohne etwas am Rechner zu ändern.

3.10 Was mache ich womit?

Ist das neue Betriebssystem auf dem Rechner, will man natürlich auch damit arbeiten. Damit die Aufgabeverteilung etwas schneller geht, kommt hier eine kleine Orientierungstabelle für die wichtigsten Arbeitsutensilien und ihre Linux-Entsprechungen, natürlich ohne Anspruch auf Vollständigkeit:

Tabelle 3.2: Programmübersicht Windows — Linux

Aktion	...unter Windows	...unter Linux
Brief schreiben, Tabellenkalkulation, Präsentationen	MS Office, MS Works	OpenOffice/StarOffice, KOffice
Grafiken und Photos bearbeiten	PhotoShop, Picture Publisher, PaintShop Pro	gimp
Diplomarbeit schreiben	MS Office	OpenOffice/StarOffice, lyx/klyx, LaTeX
Internet-surfen	Internet Explorer, Netscape, Opera, Mozilla	Netscape, Opera, Mozilla, Konqueror, Galeon
eMail	Outlook Express, Netscape Messenger, Pegasus etc.	kmail, Evolution, mutt, pine, Netscape Messenger etc.
CD's brennen	WinOnCD, Nero etc.	K3B, KOnCD, cdrecord, X-roast
MP3-hören	WinAmp etc.	xmms etc.
DVD's abspielen	PowerDVD, WinDVD etc.	xine, mplayer

3.11 Wer hilft mir, wenn was schiefgeht?

Sollte irgend etwas bei der Installation nicht so laufen, wie man es erwartet oder das System gar nicht zum Laufen zu kriegen sein, sollte man erst einmal tief durchatmen und sich freuen, daß man alle eigenen Daten vorher per Backup in Sicherheit gebracht hat. Danach ist es hilfreich, nochmal zu rekapitulieren, was man bis zum Auftreten des Problems gemacht hat. Wie es weitergeht, hängt davon ab, welche Distribution man gewählt hat. Käufer der kommerziellen Linux-Boxen von SuSE, Mandrake oder Red Hat haben natürlich Anspruch auf Installations-support, meist allerdings nur für einen Rechner und nur für das schlichte Booten des Systems bis zur grafischen Oberfläche.

Aber auch Neueinsteiger, die sich ihr Linux frei beschafft haben und daher keinen Installationssupport genießen, werden nicht im Regen stehen gelassen. Vorausgesetzt, man bekommt den Rechner noch irgendwie ans Internet, helfen einem diverse Newsforen bei der Eingrenzung des Problems und seiner Lösung. Im deutschsprachigen Raum ist `de.comp.os.unix.linux.misc` ein guter Anlaufpunkt für allgemeine Fragen. Hardware wird unter `de.comp.os.unix.linux.hardware` behandelt. Für Debian gibt es eigene News-Foren, in denen auch Debütanten geholfen wird. Für alle News-Foren gilt natürlich, am besten erst eine kleine Weile mitlesen und auch mal einen Blick in die jeweilige FAQ werfen. Viele Fragen, die man selbst hat, werden immer wieder gestellt und lassen sich durch einen Blick in die jeweilige FAQ schnell beantworten.

3.12 Links

- Linux-ISO's zum Download:
<http://www.linuxiso.org/>
- Linux-ISO's zum Bestellen:
<http://www.linux-iso.de/linux/index.html>
- SuSE Hardware-Datenbank:
<http://hardwaredb.suse.de/>
- Infos zu Linux auf Laptops:
<http://www.linux-laptop.net/>
- Knoppix, Live-CD-Linux:
<http://www.knopper.net/knoppix/>
- Linux-Unterstützung von Scannern:
<http://www.mostang.com/sane/sane-backends.html>

Kapitel 4

Kleine Baumkunde für Jungpinguine

Autorin: *Jana Jaeger*

Der erste Blick auf ein laufendes Linuxsystem befremdet den Linux-Neuling mittlerweile Dank komfortabler Desktopumgebungen wie Gnome und KDE nicht mehr so stark wie er es noch vor ein paar Jahren getan hätte. Längst kann er auch hier die meisten Dinge per Drag & Drop oder über graphische Bedienoberflächen erledigen ohne eine einzige Kommandozeile eingeben zu müssen.

Sobald er sich aber an das Innenleben seines Systems heranwagt, treten einige fundamentale Unterschiede zur vielleicht gewohnten DOS/Windows-Welt deutlich zutage. Der Linux- bzw. UNIX-Dateibaum ist nur einer von vielen.

4.1 Linux ist anders - Dateisysteme und Partitionen

Von Windows ist der Benutzer gewohnt, jeden einzelnen Datenträger (Festplatten oder auch Festplattenpartitionen, sowie die verschiedenen Laufwerke CD-ROM, Disketten, etc.) separat über einen Buchstaben C:, D:, ... anzusprechen. Unter Linux reibt er sich verwundert die Augen und wird mit folgender Nomenklatur konfrontiert:

Statt der Buchstaben findet er sogenannte Device-Namen. `/dev/hdxy` steht für eine IDE-Festplatte, wobei `<i>x</i>` die Nummer der Festplatte bezeichnet (a für die erste Platte, b für ...) und `<i>y</i>` die Partition angibt. SCSI-Festplatten werden dementsprechend mit `dev/sdxy` angesprochen. Ein kleines Beispiel:

```

jj@arthur:~ > df
Filesystem          1k-blocks      Used Available Use% Mounted on
/dev/sda3            1488740    1175312    236580  83% /
/dev/sda4             434331     377654     34246  92% /var
/dev/sdb3             537568     383756    126064  75% /home
/dev/sda2             21406       1691     18573   8% /boot

```

Das Kommando `df` liefert einen Überblick über die Organisation des Systems. In diesem Fall sind zwei SCSI-Platten `/dev/sda` und `/dev/sdb` ins System eingebunden. Die erste Partition auf der ersten Platte macht die Swappartition aus (siehe unten in der Ausgabe von `/etc/fstab`, die Bootpartition befindet sich in der zweiten Partition der ersten Platte, usw...

Zusätzlich fällt auf, daß alle Datenträger in ein- und denselben Dateibaum eingebunden sind. Unterschiedliche Festplatten werden durch unterschiedliche Pfadnamen in einem einheitlichen Dateibaum, aber nicht als unterschiedliche Laufwerke, angesprochen. Dementsprechend merkt der normale Benutzer (vorausgesetzt er nutzt eine reine Linux-Installation) nicht, daß seine Daten auf einer anderen Platte liegen. Darüber kann er sich informieren, indem er den Inhalt der Datei `/etc/fstab` untersucht.

```

/dev/sda1          swap                swap defaults    0    0
/dev/sda3          /                   ext2 defaults    1    1
/dev/sda4          /var                ext2 defaults    1    2
/dev/sdb3          /home               ext2 defaults    1    2
/dev/sda2          /boot               ext2 defaults    1    2
/dev/scd0          /cdrom              iso9660 ro,noauto,user,exec 0

```

Hier wird festgehalten, welcher Datenträger (Device) an welchem Punkt ins Dateisystem eingebunden ist (Mount point). Die dritte Spalte legt den Dateisystemtyp und gleichzeitig die Art des entsprechenden Checkprogramms für diese Partition fest. Wird beim Neustart des Systems der gesamte Datenbestand auf allen Platten und Partitionen auf Konsistenz geprüft, liest das entsprechende Checkprogramm `fsck` die Einträge in `/etc/fstab`. Besonders wichtig sind für `fsck` die Einträge in der dritten und sechsten Spalte. Swappartitionen werden nicht auf Konsistenz überprüft. Der Eintrag in der sechsten Spalte 0 weist `fsck` an, diese Partitionen zu ignorieren. Die Reihenfolge der abzuarbeitenden Dateisystemchecks legt der Eintrag in der sechsten Spalte fest. Zuerst wird die Rootpartition des Beispielsystems mit `fsck.ext2` überprüft. Anschließend werden in der Reihenfolge der Einträge (mit 2) an sechster Stelle alle folgenden Partitionen abgearbeitet. Pro Platte läuft nur jeweils ein Durchlauf des entsprechenden Checks. In diesem Fall werden `/var` auf `/dev/sda4` und `/home` auf `/dev/sdb3` parallel überprüft und am Ende folgt `/boot` auf `/dev/sda2`. Weitere Informationen zu `fstab` erhält der Benutzer mit `man fstab` oder `man mount`.

4.2 Wo liegt was? - Der Dateibaum unter Linux

Nicht nur, daß die einzelnen Datenträger innerhalb eines einzigen Dateibaumes liegen, auch alle anderen Verzeichnisse und Dateien sind streng hierarchisch einem einzigen Wurzelverzeichnis / (oder system root) untergeordnet. Diese einheitliche Sicht der Dinge hat viele Vorteile, unter anderem stellt diese Struktur sicher, daß alle Systeme gleich aufgebaut sind. Daten liegen in allen Systemen an der gleichen Stelle, egal wie viele Platten in Benutzung sind. Somit entfällt die Suche nach einzelnen Programmen oder Dateien auf den verschiedenen Platten. Es existiert ein Standard, der die für UNIX allgemein (mit speziellen Ausführungen zu Linux) geltenden Hierarchieregeln zusammenfaßt - der FHS oder Filesystem Hierarchy Standard. Abzurufen sind die neuesten Informationen zum FHS unter <http://www.pathname.com/fhs>.

Die folgende Übersicht soll grob skizzieren, wie der Dateibaum unter Linux aussieht und welche Informationen in welchen Ästen dieses Baumes verborgen liegen. Vom Wurzelverzeichnis / aus, verzweigt sich der Dateibaum in folgende Unterverzeichnisse:

4.2.1 /bin

Hier befinden sich die meisten elementaren Linux-Kommandos, z.b. Shells wie `bash`, `csch` etc. und die grundlegendsten Befehle, ohne die kein System verwaltet werden kann (`cp`, `mv`, `rm`, `cat`, `ls`). Im Gegensatz zu den Dateien in `/sbin` sind alle diese Kommandos auch vom Normalbenutzer ohne `root`-Rechte ausführbar.

4.2.2 /boot

Alles, was hier liegt, benötigt das System beim Booten (mit LILO). Oft liegt hier auch der Kernel.

4.2.3 /dev

Wie oben schon erwähnt, sieht Linux alles, aber auch wirklich alles - egal ob Platte, Partition, CD-ROM-Laufwerk, serielle Schnittstelle, ... - als Datei an. Diese speziellen Dateien werden Devices genannt und befinden sich in diesem Verzeichnis. Dementsprechend befinden sich die Daten des CD-ROM-Laufwerks in einer Datei, `/dev/cdrom`.

4.2.4 /etc

Die Konfiguration des gesamten Systems geschieht über Dateien, die hier gespeichert sind. Es ist ratsam, von diesem Verzeichnis Sicherheitskopien anzulegen.

Auf diese Weise erspart man ein mühsames Neukonfigurieren, wenn der Rechner später einmal reinstalled oder komplett neu installiert werden muß. In einem Unterverzeichnis `/X11` lagern beispielweise die Konfigurationsdaten zu X.

4.2.5 `/home`

Dieser Bereich ist die Privatsphäre des Benutzers. Da Linux als Mehrbenutzersystem ausgelegt ist, findet sich hier für jeden Benutzer ein separates Unterverzeichnis `/home/Benutzer`, auf das nur der Benutzer selbst und der Systemadministrator zugreifen können. Alle anderen Verzeichnisse sind systemweit zugänglich.

4.2.6 `/lib`

Hier liegen die Programmbibliotheken, auf die alle Programme zugreifen können (shared libraries).

4.2.7 `/lost+found`

`/lost+found` ist die Sammelstelle für alle Dateien bzw. Dateiblöcke, die bei einem unkontrollierten Herunterfahren des Systems zerstört und von `fsck` nach dem Neustart nicht wiederhergestellt bzw. richtig zugeordnet werden konnten. Meistens machen diese Daten für sich gesehen nicht mehr viel Sinn. Wenn es sich aber um eine Textdatei handelt, können die hier abgelegten Daten bei der Restauration der ursprünglichen Datei von Nutzen sein. Dieses Verzeichnis gibt es nur unter dem `ext2` Dateisystem.

4.2.8 `/mnt`

Mountpunkt, an dem der Systemadministrator `root` temporär externe Dateisysteme einbinden kann.

4.2.9 `/opt`

Unter `/opt` befindet sich alle Software und alle Zusatzpakete, die nicht im Umfang einer Standardinstallation enthalten sind. `/opt/kde` ist das Unterverzeichnis, in dem beispielsweise alle KDE-Pakete gespeichert sind. Unter `/opt/kde/bin` liegen die entsprechenden ausführbaren Programme.

4.2.10 `/proc`

Enthält Informationen über alle laufenden Prozesse und dient zur systemweiten Prozeßverwaltung. Unter `/proc` befinden sich keine echten Dateien (Zitat aus dem FHS: " ...kernel and process virtual filesystem").

4.2.11 /root

Auch der Systemadministrator/Superuser hat ein Homeverzeichnis. Auf `/root` darf nur er allein zugreifen. Hier sollten alle für die Systemadministration relevanten Dinge gespeichert werden. Dieses `root`-Verzeichnis ist nicht mit dem Wurzelverzeichnis (engl. *root*) des gesamten Systems zu verwechseln.

4.2.12 /sbin

Alle Programme, die essentiell für den Betrieb des Systems sind, lagern in `/sbin`. Anwendungen zur Systemadministration und Hardwarekonfiguration sind hier gespeichert. `lilo`, `fdisk` und `ifconfig` sind Beispiele für solche Programme, ohne die das System nicht laufen könnte. Gemeinsam ist allen hier gespeicherten Programmen, daß sie nur mit Rootrechten ausgeführt werden können.

4.2.13 /tmp

In `/tmp` legen viele Programme ihre Lockfiles an oder nutzen es zur Zwischenablage von Daten (temporäre Dateien).

4.2.14 /usr

Hier befinden sich alle Anwendungsprogramme, die der normale Benutzer im Alltag braucht (unter `/usr/bin`) und das ganze X-System. Für den Benutzer ebenfalls sehr wichtig sind die Dokumentationen zu einzelnen Programmen, die er (in einem SuSE Linux System) unter `/usr/share/doc/packages` finden kann.

4.2.15 /var

Der Name deutet es schon an, hier sind variable/veränderliche Dateien gespeichert. Hier legen Programme ihre Lockfiles in `/var/lock` (regeln Zugriffsschutz auf Devices) und Log-Dateien in `/var/log` (Protokolle aktiver Programme) an, liegen Maildateien (`/var/spool/mail`) und zwischengespeicherte Druckdateien (z.B. unter `/var/spool/lp`).

4.3 und weiter...

Bis hierhin scheint der Dateibaum noch relativ einfach nachvollziehbar aufgebaut. Die einzelnen Unterverzeichnisse können nach dem Muster der Verzweigungen unter `/` selbst in eigene Unterverzeichnisse aufgegliedert sein. So nimmt zwar die Zahl der Verzweigungen stark zu, aber gleichzeitig bleibt das System auch in tieferen Schichten noch überschaubar.

Um sich innerhalb der Verästelungen zurechtzufinden, stehen dem Benutzer mehrere Befehle zur Verfügung:

4.3.1 *which Programm*

liefert den Pfad, in dem sich das gesuchte Programm befindet

4.3.2 *locate Suchbegriff*

gibt alle Pfade aus, in denen sich der Suchbegriff befindet

4.3.3 *find Suchbegriff*

funktioniert wie `locate`, ist aber langsamer, da es nach dem Aufruf erst einmal die gesamte Platte nach dem Suchbegriff durchkämmen muß. `locate` greift auf eine in regelmäßigen Abständen erzeugte Datenbank zurück, die den Aufbau des Systems widerspiegelt, und ist dementsprechend schnell.

Kapitel 5

Dateisysteme unter Linux

Autorin: *Jana Jaeger*

Jeder nutzt sie, keiner sieht sie bei der Arbeit, und ohne sie geht **gar nichts**. Die Rede ist von den Dateisystemen. Dieses Kapitel soll einen kleinen Einblick in die Dateiwirtschaft unter Linux vermitteln und zwei der bekannteren, einen Oldie, *ext2*, und einen Newcomer, *ReiserFS*, kurz vorstellen.

5.1 Was genau tut ein Dateisystem?

Die Aufgabe eines Dateisystems besteht darin, die einzelnen Datenblöcke einer Datei auf die Festplatte (oder einen anderen Datenträger) abzulegen. Es legt fest, an welcher Stelle des Datenträgers sie gespeichert werden sollen und archiviert diese Informationen gleichzeitig in einer Tabelle. Die Zugriffs- und Lokalisierungsstrategie eines Dateisystems hat direkten Einfluß auf die Durchsatzrate der Schreib- und Lesezugriffe auf das Medium und auf die Zuverlässigkeit des Dateisystems an sich.

5.2 Welche Dateisysteme gibt es?

Unter Linux gibt es nicht **das** Dateisystem schlechthin. Linux kommt mit einer ganzen Reihe von Dateisystemen zurecht. Hier eine kurze, beileibe nicht vollständige, Übersicht der verschiedenen Dateisysteme für Linux:

AFS ein verteiltes Dateisystem, meist im WAN genutzt

autofs für das automatische Mounten eines Dateisystems bzw. Datenträgers ins Gesamtsystem - ist also kein echtes Dateisystem

devpts für Pseudoterminale (nach UNIX-98-Spezifikation)

- ext2* der Standard unter Linux
- ext3* Weiterentwicklung von *ext2* (noch Beta Status), wird Journaling beherrschen
- hpfs* das OS/2-Dateisystem
- iso9660* Standardformat für CD-ROMs und DVDs
- JFS* Von IBM entwickeltes Journaling Filesystem, ursprünglich für AIX (IBMs UNIX) entwickelt und seit Februar 2000 ist eine Beta-Version unter einer Open Source Lizenz auch als Patch für Linux verfügbar
- minix* Dateisystem von Minix, wird oft für Linux-Disketten gebraucht.
- msdos* Dateisystem für DOS-Partitionen und -Disketten (kurze Dateinamen)
 - nfs* Netzwerkdateisystem für UNIX, um Daten von anderen Rechnern zu lesen
 - ntfs* Windows-NT/2000 Dateisystem (allerdings nur mit Lesezugriff)
 - nwfs* kann ein NetWarediskfilesystem lesen
 - proc* Prozeßverwaltung unter `/proc`
- ReiserFS* Reiser-Dateisystem, nach seinem Schöpfer Hans *Reiser* benannt
 - mbfs* Samba (Netzwerkdateisystem unter Windows)
 - swap* Swap-Partitionen oder -Dateien
 - udf* Universal Disk Format (CD-RWs und DVDs)
- usbdevfs* Einbinden und Verwalten von USB-Geräten
 - vfat* DOS/Windows-9x-Dateisystem (lange Dateinamen)
- XFS* Journaling Filesystem unter UNIX. Ursprünglich von SGI für IRIX, das hauseigene UNIX entwickelt, aber mittlerweile unter der GPL in einer Beta-Version als Patch für Linux verfügbar

5.3 Zwei aus X

Aus der Fülle der oben genannten Beispiele werden nun zwei herausgegriffen, um die unterschiedlichen Lösungsansätze für das Speicherproblem unter Linux zu diskutieren.

5.3.1 ext2

ext2 bewältigt seine Verwaltungsaufgabe, indem es den vorhandenen Speicherplatz in Blöcke gleicher Größe (z.B. je 1024 Byte) einteilt und diese Blöcke durchnummeriert. Die Menge der Blöcke wird anschließend in verschiedene Gruppen aufgeteilt, die jeweils zur Speicherung unterschiedlicher Datentypen genutzt werden.

Von "Gruppe" kann man bei der ersten Gruppe in der Reihe eigentlich nicht sprechen. Es handelt sich hierbei um einen einzelnen Block, den Bootblock. Er enthält eine Menge von Minimaldaten, die zum Start des Betriebssystems notwendig sind. Die zweite "Gruppe" umfaßt ebenfalls nur einen einzigen Block, der Superblock genannt wird. Im Superblock wird festgelegt, wie groß die nun folgenden Gruppen sind. Er enthält darüber hinaus Informationen über das gesamte Filesystem. Aus Sicherheitsgründen wird der Superblock repliziert. Jetzt endlich folgen wirkliche "Gruppen" von Blöcken. Die dritte und die vierte Gruppe fungieren als eine Art von Landkarten (Bitmaps) über die beiden letzten Gruppen. Die Blöcke der dritten Gruppe sind als Inode Bitmap organisiert. Hier wird protokolliert, welche Blöcke für Inodes frei sind. In der vierten Gruppe findet sich eine entsprechende Bitmap über die verfügbaren Blöcke für Daten. In der fünften Gruppe liegt der eigentliche Speicherplatz für Inodes (je acht Inodes können bei einer Blockgröße von 1024 Bytes in einem Block gespeichert werden). Die sechste Gruppe dient als Speicherplatz für Daten (1024 Bytes pro Block). Hier und nur hier werden die eigentlichen Daten gespeichert, die anderen fünf Gruppen sind der Verwaltungsüberbau.

Wichtigste Größe in einem UNIX-Filesystem und damit auch in *ext2* sind die Inodes (oder auch Informationsknoten), die mit Ausnahme des Dateinamens alle für eine Datei relevanten Daten enthalten. So werden hier gespeichert:

- Benutzer- und Gruppen-ID
- Zugriffsrechte
- Größe der Datei
- Anzahl der Links, die auf diese Datei verweisen
- Daten der Erstellung, letzten Änderung oder Löschung dieser Datei
- Verweise auf die ersten zwölf Blöcke dieser Datei
- wenn nötig: Ein-, zwei- oder gar dreifach indirekte Verweise auf weitere Datenblöcke (dazu später mehr ...)

Eine Datei besteht also aus ihrem entsprechenden Inode und aus mehreren Datenblöcken, die die eigentlichen Daten enthalten. Die Verwaltungsinformationen im Inode und der eigentliche Dateninhalt der Datei sind vollkommen getrennt

und können auch unabhängig voneinander bearbeitet werden. Wer eine Datei in ein neues Verzeichnis verschiebt, ändert den Inode und die entsprechenden Verzeichniseinträge, aber niemals wird er die Daten direkt ändern.

Umfaßt eine Datei mehr als 12 kByte (12 mal 1024 Byte oder zwölf Datenblöcke), birgt ein weiterer Inode im Inode einen Verweis auf einen Block, der selbst keine Daten, dafür aber Verweise auf bis zu 256 weitere Datenblöcke enthält. Hier spricht man vom einfach indirekten Verweis. Bei Dateien einer über einer Größe von 268 kByte (12 Blöcke direkt referenziert, 256 einfach indirekt) verweist ein weiter Inode auf einen Block, der auch selbst wieder auf 256 Datenblöcke zeigt, die wiederum selbst auf 256 Datenblöcke zeigen können. In diesem Fall redet man vom zweifach indirekten Verweis. Natürlich läßt sich auch ein dreifach indirekter Verweis realisieren.

Die Verwaltung von Dateien und Verzeichnissen ist sich sehr ähnlich. Ein Verzeichnis ist im Prinzip wie eine Textdatei mit mehreren Zeilen gegliedert. In jeder Zeile dieser Textdatei steht der Name eines Unterverzeichnisses oder einer Datei und die Inode-Nummer der Datei. Zusätzlich findet sich hier ein Verweis auf sich selbst (.) und das übergeordnete Verzeichnis (..).

Die Dateiverwaltung läuft nun, wie an der folgenden Ausgabe des Kommandos `ls -lai` zu sehen ist, ab:

```
jj@gromit:/tmp/tmp/jj > ls -lai
insgesamt 4
223244 drwxr-xr-x    4 jj users      1024 Jan 21 17:00 .
180247 drwxr-xr-x    3 jj users      1024 Jan 21 17:03 ..
 84002 drwxr-xr-x    2 jj users      1024 Jan 21 17:00 dir1
223245 drwxr-xr-x    2 jj users      1024 Jan 21 17:01 dir2
```

Von links nach rechts erklärt: Die linke Spalte gibt die Inode-Nummern an, unter der die Verweise, Unterverzeichnisse oder gegebenenfalls auch Dateien abgelegt sind. In der nächsten Spalte werden die Zugriffsrechte aufgelistet. Darauf folgt die Spalte mit der Anzahl fester Links, die auf diese Dateien oder Verzeichnisse zeigen (Hardlinks). Eine Datei hat zwar nur einen Inode, aber in diesem können mehrere Links auf diese Datei gespeichert werden. Beispielsweise ist für das Unterverzeichnis `dir1` die Zahl von zwei Hardlinks angegeben. Konkret besagt dies: `dir1` verweist einmal auf sich selbst und enthält einen Verweis auf das Verzeichnis eine Ebene darüber, hier `jj`. Die nächsten Spalten geben Benutzer- und Gruppen-ID, Verzeichnisgröße, Änderungsdatum und Namen an.

Jetzt zum Mounten und Unmounten der entsprechenden Platte oder Partition und den Geschehnissen nach einem Absturz bzw. einem "sauberen" Herunterfahren des Systems. Im Superblock (Gruppe 2) wird beim Mounten des Dateisystems ein sogenanntes Valid Bit gelöscht, das nach einem sauberen Unmount wieder gesetzt wird. Fehlt dieses Bit beim nächsten Start des Systems, bedeutet dies, daß das System nicht ordnungsgemäß heruntergefahren wurde. Jetzt wird ein Programm gestartet, *e2fsck*, das die Daten auf Konsistenz überprüft und wenn nötig

zu reparieren versucht. Dies ist nicht immer erfolgreich, Dateien (oder Datenblöcke), die nicht ordnungsgemäß wiederhergestellt werden konnten oder nicht richtig referenziert werden können, landen in einem extra dafür vorgesehenen `/lost+found`-Verzeichnis der Partition.

Da `fsck` über die gesamte Partition (oder Platte) laufen muß, kann diese Prozedur unter Umständen je nach Größe des Dateisystems einige Minuten kostbarer Uptime in Anspruch nehmen. Ist man auf eine effizientere (schnellere) Art der Dateiverwaltung angewiesen, sollte man eine andere Art der Dateiverwaltung in Betracht ziehen, z. B.:

5.3.2 *ReiserFS* - ein Filesystem mit Journaling

ReiserFS ist ein hybrides Filesystem, d.h. es wird ein "normales", nicht log-orientiertes Filesystem erstellt, für das es noch zusätzlich ein Journal gibt. In diesem Journal werden alle Metadatenänderungen, die mehr als einen Block betreffen, vorgemerkt, und erst wenn sie im Journal stehen in das echte Filesystem geschrieben. Daten werden direkt in das Dateisystem geschrieben. Bei *ReiserFS* kann es also weiterhin zu einem Datenverlust bei einem Systemabsturz kommen, nur die Metadaten sind in Ordnung.

Beim Mounten einer *ReiserFS* Partition wird die Log-Datei ausgewertet und die Änderungen ausgeführt, die aufgrund eines Systemabsturzes nicht durchgeführt werden konnten. Die Partition ist dann wieder in einem konsistenten Zustand. Diese einfache Rekonstruktion des Status Quo hat einen großen Vorteil: Der Filesystem-Check beim Booten eines Systems dauert nur wenige Sekunden, auch bei großen Platten wo *ext2* mehrere Minuten benötigen würde.

Natürlich wird man die Vorteile des "Journaling" nicht genießen können, ohne auch ein paar Nachteile in Kauf zu nehmen. Die Schreibzugriffe auf die Platte sind häufiger, da Änderungen erst angemeldet und dann durchgeführt werden.

Das *ReiserFS* besitzt aber eine Reihe von Features, die es vor allem bei kleinen Dateien und vielen Dateien in einem Verzeichnis schneller macht als *ext2*. Die zentrale Datenstruktur bei *ReiserFS* ist ein Baum, Informatiker bezeichnen die implementierte Variante als einen B*-Baum. In den Blättern des Baumes werden vier verschiedene Arten von Datensätzen gespeichert:

direkte Datensätze:

Diejenigen Teile von Dateien, die nicht mehr in einen Block reinpassen. Der Plattenplatz wird dadurch besser ausgenutzt (ein 10 Zeichen gro Datei landet auf einem traditionellen System wie *ext2* in einem Block von z.B. 1024 Bytes) und kleine Dateien sind direkt im Baum, es ist kein zusätzlicher Plattenzugriff notwendig.

indirekte Datensätze:

Dies sind Zeiger auf große Dateien, die außerhalb des Baumes liegen. Alle Daten,

bis auf den letzten Teil (siehe direkte Datensätze) sind in solchen Dateien gespeichert.

Verzeichnisse:

Enthält die einzelnen Einträge eines Verzeichnisses.

”stat data”:

ext2 benutzt Inodes, *ReiserFS* speichert dagegen diese Daten im Baum ab und belegt somit nicht unnötige Inodes (bei *ext2* wird die Anzahl und Lage der Inodes bei der Formatierung festgelegt und kann später nicht geändert werden).

Jeder Datensatz besitzt einen eindeutigen Schlüssel (hier werden Hashfunktionen benutzt), der zum Sortieren und Wiederfinden benutzt wird.

Das Verwalten des B*-Baumes kostet etwas mehr Zeit als die simplen Techniken von *ext2*, aber dafür sind Zugriffe effizienter durchführbar. Je nach Benchmark kann *ext2* oder *ReiserFS* schneller sein, aber im normalen Einsatz überwiegen die Vorteile von *ReiserFS* deutlich.

Kapitel 6

Innenansichten eines Linuxsystems

Autoren: *Stephan Barth, Jana Jaeger*

Wann immer Probleme mit Ihrem Linuxsystem auftreten, und Sie den Dingen auf den Grund gehen oder Ihre Erfahrungen mit jemandem teilen wollen, benötigen Sie ein paar grundlegende Handwerkszeuge zum Aufspüren von Hard- und Softwareinformationen Ihres Systems. Die folgende Liste gibt einen kleinen Überblick über die wichtigsten Befehle. Natürlich gibt es noch viele andere Befehle, die das Innenleben eines Linuxsystems beleuchten.

6.1 lspci

`lspci` listet alle Komponenten des PCI-Busses auf. Mit der Option `-vv` erhalten Sie eine ausführliche Liste, mit `-vn` nur die IDs der Geräte. Bitte denken Sie daran, diesen Befehl als `root` auszuführen.

6.2 grep

`grep` ist der gebräuchlichste Befehl zum Durchsuchen von Dateien unter Unix/Linux. Wichtig: standardmäßig beachtet der Befehl die Klein- und Großschreibung. Mit `-i` kann dies abgeschaltet werden.

Beispiel:

Ohne `-i`:

```
linux:~ # lspci | grep controller
00:06.0 SCSI storage controller: Adaptec AHA-2940U2/W / 7890
00:09.0 Ethernet controller: Digital Equipment Corporation DECchip 21140 [FasterNet] (rev 22)
00:0a.0 Multimedia audio controller: Ensoniq ES1371 [AudioPCI-97] (rev 06)
01:00.0 VGA compatible controller: nVidia Corporation Riva TnT 128 [NV04] (rev 04)
```

und mit `-i`. Hier wird der USB Controller noch mit aufgelistet.

```
linux:~ # lspci | grep -i controller
00:04.2 USB Controller: Intel Corporation 82371AB PIIIX4 USB (rev 01)
00:06.0 SCSI storage controller: Adaptec AHA-2940U2/W / 7890
00:09.0 Ethernet controller: Digital Equipment Corporation DECchip 21140 [FasterNet] (rev 22)
00:0a.0 Multimedia audio controller: Ensoniq ES1371 [AudioPCI-97] (rev 06)
01:00.0 VGA compatible controller: nVidia Corporation Riva TnT 128 [NV04] (rev 04)
```

Das Zeichen `|` übergibt die Ausgabe des ersten Befehls an den zweiten. Das spart einen Schritt. Sonst müßte die Ausgabe von `lspci` in eine Datei umgeleitet (mit `>`) und diese dann von `grep` ausgelesen werden:

```
linux:~ # lspci > datei
linux:~ # grep controller datei
```

Die Ausgabe von `grep` wurde absichtlich nicht noch einmal aufgeführt. `grep` kann auch rekursiv arbeiten, d.h. auch gleich in Unterverzeichnissen mit suchen. Das geht mit dem Parameter `-r`.

Beispiel für die Suche nach einem Treiber für eine spezielle Netzwerkkarte in den Quellen des Kernels:

```
linux:/usr/src/linux/drivers/net # grep -r Tulip *
```

Hier ist die Ausgabe zu umfangreich gewesen, deswegen ist sie hier ebenfalls nicht eingefügt.

6.3 RPM

Praktisch alle Dateien, die auf einem SuSE Linux System zu finden sind, werden aus RPM-Paketen installiert und sind in der systemweiten RPM-Datenbank registriert. Ausnahmen sind vereinzelte, generierte Konfigurationsdateien oder die Dateien des `/proc`-Dateisystems, welche nur virtuell sind und zur Laufzeit vom Kernel erzeugt werden.

Wenn man wissen will, ob ein Paket installiert ist, kann man das über den Query-Parameter `-q` machen:

```
linux:~ # rpm -q rpm
rpm-3.0.6-26
```

Hierbei wird auch gleich die Version des Programms an sich ausgegeben (3.0.6) und die SuSE-Version von dieser Programmversion (26). Eine neuere SuSE-Version kann z.B. aktuellere Patches enthalten als die reine Programmversion.

Um die Beschreibung und die Dateiliste des Pakets zu erhalten, gibt es die zusätzlichen Parameter `i` und `l`:

Die Paketbeschreibung von YaST2:

```

linux:~ # rpm -qi yast2
Name       : yast2                      Relocations: (not relocateable)
Version    : 2.1.151                   Vendor: SuSE GmbH, Nuernberg, Germany
Release    : 0                         Build Date: Mon Jan 22 18:28:26 2001
Install date: Fri Feb  2 09:53:10 2001 Build Host: Flugge.suse.de
Group      : System Environment/YaST   Source RPM: yast2-2.1.151-0.src.rpm
Size       : 1717027                   License: (c) 2000 SuSE GmbH
Packager   : feedback@suse.de
Summary    : YaST2 - installation scripts and documentation
Description:
This package contains scripts and data needed for SuSE linux
installation with YaST2

Authors:
-----
...

SuSE series: a

```

Der Paketinhalt von Pine:

```

linux:~ # rpm -ql pine
/etc/pine.conf
/etc/pine.conf.fixed
/usr/bin/mttest
/usr/bin/pine
/usr/share/doc/packages/pine
/usr/share/doc/packages/pine/COPYRIGHT
/usr/share/doc/packages/pine/brochure.txt
/usr/share/doc/packages/pine/mailcap.unx
/usr/share/doc/packages/pine/tech-notes
/usr/share/doc/packages/pine/tech-notes.txt
/usr/share/doc/packages/pine/tech-notes/Makefile
/usr/share/doc/packages/pine/tech-notes/background.html
/usr/share/doc/packages/pine/tech-notes/cmd-line.html
/usr/share/doc/packages/pine/tech-notes/config-notes.html
/usr/share/doc/packages/pine/tech-notes/config.html
/usr/share/doc/packages/pine/tech-notes/for.pnuts
/usr/share/doc/packages/pine/tech-notes/index.html
/usr/share/doc/packages/pine/tech-notes/installation.html
/usr/share/doc/packages/pine/tech-notes/introduction.html
/usr/share/doc/packages/pine/tech-notes/low-level.html
/usr/share/doc/packages/pine/tech-notes/porting.html
/usr/share/man/man1/pine.1.gz

```

Will man wissen, ob und wo sich eine Datei auf dem System befindet, ist `locate` der Befehl der Wahl. In `/lib` liegen die aktuellen Systembibliotheken. U.a. in `/usr/i486-linux-libc6/lib` Kompatibilitätsbibliotheken für ältere Programme.

```
linux:~ # locate libc.so.6
/lib/libc.so.6
/usr/i486-linux-libc6/lib/libc.so.6
```

Umgekehrt kann man auch herausfinden, zu welchem Paket eine Datei gehört. Dies geschieht mit dem Parameter `f`:

```
linux:~ # rpm -qf /lib/libc.so.6
glibc-2.2-7
```

Falls man von einem Paket lediglich eine Liste der beiliegenden Dokumentation sehen will, gibt es dafür auch einen Befehl:

```
linux:~ # rpm -qd apache
```

Will man ein z.B. heruntergeladenes Paket vor dem Installieren inspizieren, geht das am einfachsten mit dem Befehl `less`:

```
less kinternet.rpm
```

Alternativ geht das auch mit den RPM-Optionen `-qip` (Paketinformation) und `-qlp` (Paketliste). Das `p` sagt dem RPM-Befehl, daß es nicht in der Paketdatenbank nachsehen soll, sondern direkt in der angegebenen Datei.

```
rpm -qlp xmms.rpm
rpm -qip xmms.rpm
```

6.4 Informationen über nicht installierte Pakete

Allerdings helfen all die schönen RPM-Befehle wenig, wenn ein Paket nicht auf dem System installiert ist. Dafür können Sie in der Datei

```
ARCHIVES.gz
```

von der ersten SuSE CD mit dem Befehl

```
zgrep
```

(das `z` bei `zgrep` bedeutet, daß dieser Befehl `.gz` Dateien durchsuchen kann) suchen.

```
linux:~ # zgrep /usr/bin/pine ARCHIVES.gz
./suse/n1/pine.rpm: -rwxr-xr-x root root 2967892 Mar 1 13:20 /usr/bin/pine
```

6.5 Weitere Hilfen

Mit diesem Bündel an Befehlen verschaffen Sie sich (und anderen, z.B. den Mitarbeitern im Installationssupport oder den Lesern einer Mailingliste) die grundlegendsten Informationen über Ihr System.

Sollten Sie zusätzlich Informationen über den Systemzustand benötigen, schauen Sie sich die Ausgaben der folgenden Befehle an:

6.5.1 df

Dieser Befehl gibt Auskunft über den verfügbaren und benutzten Plattenplatz. `-h` macht die Ausgabe ein wenig verständlicher.

6.5.2 du

Mit `du` erhalten Sie - am besten als Ergänzung zu einem vorher erfolgten Aufruf von `df` - den von Unterverzeichnissen und einzelnen Dateien belegten Platz. Besonders vor Aufräumaktionen ist dies ein nützlicher Befehl, um den Entscheidungsprozess zu beschleunigen ;-). Mit der Option `-h` von `human readable` wird die Ausgabe verständlich.

6.5.3 free

Sollten Sie Auskunft über die Auslastung des Arbeitsspeichers und des Swap-Speichers benötigen. Voila:

```
\end{scriptsize}
      total      used      free      shared  buffers  cached
Mem:    196184    151572    44612         0     14416    65928
-/+ buffers/cache:    71228    124956
Swap:    136544         0    136544
```

6.5.4 ps

`ps` ohne weitere Argumente gestartet, liefert alle vom Benutzer gestarteten Prozesse. Die von anderen Benutzern auf dem aktuellen Rechner gestarteten Prozesse werden durch Angabe der zusätzlichen Option `-a` ausgegeben. `!b;Alle!;b;` Prozesse (auch solche, die von keinem Terminalfenster aus kontrolliert werden), geben Sie mit `ps -xa` aus.

6.5.5 pstree

Wie der Name schon nahelegt, gibt dieser Befehl eine grafische Übersicht über die laufenden Prozesse:

```

init--atd
|-cron
|-in.identd---in.identd---3*[in.identd]
|-inetd
|-8*[kdeinit]
|-kdeinit--3*[kdeinit]
|       |-kdeinit--bash---pine
|       |       '-bash---pstree
|       |-netscape---netscape
|       '-xemacs---gnuserv
|-kdeinit---cat
|-kdm--X
|       '-kdm---kde---ksmserver
|-kflushd
|-khubd
|-klogd
|-knotify
|-kreiserfsd
|-kswapd
|-kupdate
|-lpd
|-mdrecoveryd
|-6*[mingetty]
|-nscd---nscd---5*[nscd]
|-rpciod
|-sshd
|-syslogd
|-usbmgr
|-xconsole
'-ypbind---ypbind---2*[ypbind]

```

Mehr Informationen über die einzelnen genannten Befehle erhalten Sie mit einem der wichtigsten Linux/Unix-Befehle überhaupt: `man Befehl`. Die sogenannten Manpages enthalten stark komprimierte Beschreibungen der meisten Befehle. Natürlich gibt es auch `man man`

Kapitel 7

Wichtige Befehle für Umsteiger

Autor: *Martin Sommer*

Nachdem im Kapitel "Innenansichten eines Linuxsystems" einige wichtige Befehle vorgestellt wurden, mit denen man sich Informationen über sein System, seine Geräte, seinen Speicher, über laufende Prozesse oder installierte Programme anzeigen lassen konnte, werden nun Befehle erläutert, die vor allem den Umgang mit Dateien und Verzeichnissen betreffen. Auch wenn heute viele dieser Befehle mit modernen Dateimanagern wie z.B. dem *Konqueror* oder dem *Midnight Commander*, komfortabel zu erledigen sind, kann es doch auch von Vorteil sein, die Befehle zu kennen, um solche Prozeduren quasi von Hand zu erledigen, zumal die Eingabe von Befehlen oft wesentlich schneller ist, als das umständliche Klicken mit verschiedenen Maustasten. Zugegeben, der *Midnight Commander* ist beim Dateihandling, was kopieren, verschieben oder anzeigen angeht, durch seine Bedienung per F-Tasten an Geschwindigkeit nicht zu schlagen. Jedoch gibt es einige Operationen, die auch mit *mc* nicht durchführbar sind oder länger dauern als per Befehlseingabe.

Dieses Kapitel beschreibt, wie man Verzeichnisse anlegt und sich auf verschiedene Weisen anzeigen läßt. Es erläutert, wie man Dateien und Verzeichnisse kopiert und verschiebt. In diesem Zusammenhang wird auch ein Befehl beschrieben, mit dem man Dateien von der Workstation zu einem Server und umgekehrt kopiert, was vor allem für Leute, die private Homepages betreiben, von Bedeutung sein kann. Weiter wird darauf eingegangen, wie Eigentümer und Rechte für Dateien und Verzeichnisse vergeben werden. Zum Schluß wird erklärt, wie abgestürzte Programme (sprich hängengebliebene Prozesse) beendet werden, ohne daß der Rechner neu gebootet werden muß.

7.1 Umgang mit Dateien und Verzeichnissen

Vorbemerkung: In SuSE Linux sind zur Vereinfachung für Benutzer, die von Windows umgestiegen sind, einige Befehle auch durch die aus Windows bekannten Eingaben zu benutzen. Hierfür sind sogenannte Aliase gesetzt, die man auch selbst verändern kann. Sie stehen normalerweise in der Datei `/etc/profile`. Bei den im folgenden aufgeführten Befehle, werden, wenn Alias-Namen existieren, diese angegeben. Bei allen Befehlen wird immer nur eine kleine Auswahl an möglichen Optionen beschrieben. Ausführliche Listen über alle jeweils verfügbaren Optionen erhält man über die Manual-Seiten, die durch die Eingabe `man Befehl` angezeigt werden.

7.1.1 mkdir - Verzeichnis anlegen

`mkdir Verzeichnisname` legt ein neues Unterverzeichnis in dem Verzeichnis an, in dem Sie sich gerade befinden.

Alias: `md`. Der Befehl `md` entspricht hier der Eingabe `mkdir -p`. Durch die Option `-p` wird keine Fehlermeldung ausgegeben, wenn das Verzeichnis schon existiert.

7.1.2 ls - Verzeichnisinhalt anzeigen

`ls` ohne Optionen zeigt alle Dateien (außer den versteckten, die mit einem `.` beginnen), symbolische Links und Unterverzeichnisse des aktuellen Verzeichnisses an. Verschiedene Dateitypen, Verzeichnisse und Links werden dabei farblich unterschiedlich dargestellt. Z.B. sind normale Dateien schwarz, ausführbare rot, gepackte dunkelrot, Bilder rosa, Verzeichnisse blau und symbolische Links schwarz und fett.

Wichtige Optionen sind z.B. `l`, `a` und `F`. Die Befehle lauten also `ls -l`, `ls -a` und `ls -F` oder in Kombination, z.B. `ls -laF`. `ls -l` bewirkt das Anzeigen der langen Dateiliste. An einer Beispielausgabe werden die Spalten von links nach rechts erläutert:

```
ich@meinrechner: ls -l /etc/vmware
-r--r--r--  1 root  root    314 Jul 12 09:59 config
-r-xr-xr-x  1 root  root   9971 Jun  5 21:23 installer.sh
-rw-r--r--  1 root  root  28557 Jul 12 09:59 locations
-rw-r--r--  1 root  root     0 Jul 12 09:59 not_configured
drwxr-xr-x  4 root  root   4096 Jul 10 12:08 vmnet1
```

Die erste Spalte gibt den Typ (Verzeichnis, Datei oder Link) und die Rechte an (dazu unten mehr), dann folgt die Zahl der Links auf das jeweilige Objekt. Jetzt folgen nacheinander: Eigentümer, Gruppe, Dateigröße, Datum der letzten

Änderung, Name. Die Option `-a` zeigt zusätzlich alle versteckten Dateien an, also diejenigen, die mit einem `.` beginnen. Die Option `-F` verdeutlicht den Typ der Objekte zusätzlich zur Farbgebung noch mit einem Zeichen am Ende (z.B. `*` für eine ausführbare Datei oder `/` für ein Verzeichnis. Zu `ls` mit den genannten drei Optionen existieren einfache Alias-Namen, wobei die einfachste (1) `ls -aLF` ausführt. Hier ein Auszug aus `/etc/profile` zu den Alias-Befehlen von `ls`:

```
alias dir='ls -l'  
alias ll='ls -l'  
alias la='ls -la'  
alias l='ls -aLF'  
alias ls-l='ls -l'
```

Sie sehen, nicht nur `ll` bewirkt `ls -l`, sondern auch der alte DOS-Befehl `dir`. Sie können also auch mehrere Aliasnamen für denselben Befehl definieren.

7.1.3 cd - Verzeichnis wechseln

`cd Verzeichnis` wechselt in das angegebene Verzeichnis. Hier können Sie absolute und relative Pfade angeben. Wenn Sie sich in Ihrem `home`-Verzeichnis befinden, bewirkt `cd bilder` einen Wechsel in das Unterverzeichnis `bilder`. `cd ../../usr/local` wechselt von Ihrem `home` nach `/usr/local`, also erst zwei Hierarchiestufen hoch und dann von `/` aus nach `usr/local`. Das gleiche erhalten Sie von jeder Stelle des Dateibaumes aus mit `cd /usr/local`, also mit der absoluten Pfadangabe. Eine Stufe in der Dateihierarchie nach oben geschieht durch `cd ..`. Beachten Sie (im Gegensatz zu Windows) hier das Leerzeichen zwischen `..` und `cd`. Auch hierfür gibt es Aliasnamen in `/etc/profile`:

```
alias ..='cd ..'  
alias ...='cd ../../..'
```

Das bedeutet, `..` bringt Sie ein Verzeichnis höher und `...` führt Sie zwei Stufen nach oben. Die reine Eingabe von `cd` führt sie von jeder Stelle des Verzeichnisbaums in Ihr `home` zurück. Gleiches bewirkt `cd .`.

7.1.4 cp - Dateien und Verzeichnisse kopieren

`cp Quelle Ziel` kopiert Dateien von einem Quellort zu einem Zielort. Dieser kann im gleichen Verzeichnis in einem neuen Namen bestehen oder in einem anderem Verzeichnis unter einem beliebigen Namen. Auch hier können absolute und relative Pfadnamen eingegeben werden. Die wichtigste Option ist hier sicher `-R`. Damit kopieren sie rekursiv, also mit allen Unterverzeichnissen und deren Inhalten.

7.1.5 mv - Dateien und Verzeichnisse verschieben oder umbenennen

`mv` *Quelle Ziel* arbeitet analog dem `copy`-Befehl (`cp`), legt jedoch keine Kopien an. D.h. die Datei erscheint an dem Zielort, die Ursprungsdatei wird jedoch gelöscht. Umbenennen in demselben Verzeichnis bedeutet dann einfach, die Datei im gleichen Verzeichnis zu einer Datei mit gleichem Inhalt aber anderem Namen hin zu verschieben.

7.1.6 rm - Dateien löschen

`rm` (remove) wird benutzt wie unter DOS der Befehl `del` (delete). Es gibt einen entscheidenden Unterschied zu DOS-basierten Systemen: Das Löschen unter Linux ist (aus Sicherheitsgründen) **nicht** rückgängig zu machen. D.h. es gibt kein `undelete` oder `unrm` oder dergleichen. Sie müssen also diesen Befehl sehr vorsichtig verwenden. Auch hier können Sie wie bei `cp` mit der Option `-R` rekursiv arbeiten. Alternativ kann hier auch `-r` verwendet werden.

7.1.7 rmdir - Verzeichnisse löschen

`rmdir` *Verzeichnis* löscht das angegebene Verzeichnis. Als Alias können Sie hier `rd` verwenden.

7.1.8 ln -s - Symbolischen Link anlegen

Eine sehr praktische Funktion ist das Arbeiten mit symbolischen Links. Nehmen wir an, Sie haben Verzeichnisse oder Dateien, die Sie oft verwenden, die aber aufgrund der Datei- und Verzeichnisstruktur von Linux nicht in Ihrem Homeverzeichnis liegen können, sondern z.B., falls der Apache Webserver auf Ihrem System läuft, in `/usr/local/httpd/htdocs`. Sie möchten aber nicht jedesmal diesen langen Pfad angeben (den man sich in manchen Fällen vielleicht nur schwer einprägen kann), wenn Sie von Ihrem Homeverzeichnis aus zu den jeweiligen Daten wollen. Um dieses Problem zu lösen, legen Sie in Ihrem Homeverzeichnis einen symbolischen Link, auch weicher Link oder Softlink genannt. Sie wechseln dann zu diesem Link, als wäre er ein Verzeichnis und befinden sich danach in dem Verzeichnis, auf das der Link verwiesen hat. Um z.B. das Verzeichnis `/usr/local` in Ihrem Homeverzeichnis unter dem Namen `local` erreichen zu können, geben Sie folgenden Befehl ein (wenn Sie sich in Ihrem Home befinden):

```
ln -s /usr/local local
```

Das schon existierende Verzeichnis wird also als erstes genannt. Schauen Sie sich nun mit `1` Ihr Homeverzeichnis an. Sie sehen dann in etwa folgende Zeile:

```
lrwxrwxrwx  1 User  Gruppe   35 Jul 16 14:48 local -> /usr/local/
```

Wenn Sie nun `cd local` eingeben, landen Sie im Verzeichnis `/usr/local/` obwohl es so aussieht, als wären Sie im Verzeichnis `local`. Der erste Buchstabe der Ausgabezeile oben lautet hier `l`. Daran sehen Sie, daß es sich um einen Link handelt. Ein Verzeichnis hat hier ein `d`, eine Datei einen Strich (`-`).

7.2 Kopieren zwischen Workstation und Server oder zwischen zwei Servern

7.2.1 scp - Secure Copy

Einige Unix/Linux-Befehle haben sichere Pendanten, um den kritischeren Datenaustausch zwischen verschiedenen Computern zu bewerkstelligen. Dies ist v.a. im Serverbereich von Bedeutung, v.a. wenn auf Server zugegriffen wird, die z.B. am Internet hängen und somit vielfach Angriffsmöglichkeiten ausgesetzt sind. Wenn Sie Zugang zu einem solchen Server haben, können Sie normalerweise auch über das bekannte `ftp`-Protokoll auf die Daten zugreifen. Eine wesentlich sicherere Variante ist jedoch `scp`. Falls Sie also eine private Webseite betreiben, können Sie per `scp` anstatt per `ftp` Daten sowohl hoch-, als auch herunterladen. Nehmen wir an, Sie haben als User Hans Zugriff auf den Webserver mit der Adresse `www.hansilein.de`. Sie wollen nun alle Dateien aus Ihrem Verzeichnis `/home/webseiten/bilder/` auf Ihren Webserver ins Verzeichnis `/htdocs/hansi/images` kopieren. Sie wechseln dazu in das Verzeichnis auf Ihrem Rechner:

```
cd ~/webseiten/bilder
```

und geben dann ein:

```
scp ./* Hans@www.hansilein.de:/htdocs/hansi/images
```

`./*` bedeutet *j*alle Dateien in dem Verzeichnis, in dem ich mich befinde*j*. Der `*` in Linux als Wildcard (Platzhalter) bedeutet das selbe wie in DOS `*.*`. Der Befehl `scp` kann auch verwendet werden, um auf dem gleichen Rechner Dateien zwischen zwei verschiedenen Benutzern zu kopieren.

7.3 Eigentümer und Rechte vergeben

7.3.1 chown - Eigentümer ändern

Zum Ausführen des `chown` Befehls (change owner) müssen Sie sich als User `root` anmelden. Damit können Sie den Eigentümer von Dateien und Verzeichnissen

sowie die Gruppenzugehörigkeit ändern. Die Syntax ist einfach und auch hier funktioniert das rekursive Verfahren wieder mit der Option `-R`. Wir nehmen an, Sie haben sich in `/usr/local/` das Verzeichnis `hansidaten` angelegt, daß sie als normaler User benutzen wollen, nicht als `root`. Da sie dieses Verzeichnis aber als `root` anlegen mußten, weil Sie als normaler User in `/usr/local` lediglich Lese-rechte hatten, gehört dieses Verzeichnis erst einmal dem User `root` in der Gruppe `root`. Damit es dem User `hansi` aus der Gruppe `benutzer` zugewiesen wird, geben Sie folgenden Befehl ein:

```
chown hansi.benutzer /usr/local/hansidaten
```

oder alternativ

```
chown hansi:benutzer /usr/local/hansidaten
```

7.3.2 `chmod` - Rechte ändern

Der Befehl `chmod` (change mode) ändert die Rechte an Dateien und Verzeichnissen. Hierzu sollte man als Umsteiger von Windows etwas über Unix-Rechte wissen. Wie oben erwähnt, gehört jede Datei und jedes Verzeichnis einem Besitzer und dieser wiederum zu einer Gruppe. Rechte können nun für den Besitzer, für die Gruppe und für den Rest der Welt separat gesetzt werden. Jedes Objekt kann für diese drei (Besitzer, Gruppe und Rest) lesbar, beschreibbar und ausführbar sein. Hier kommt nun die etwas kryptische Buchstabenfolge als der `ls -l`-Ausgabe ins Spiel, die schon gezeigt wurde. Legen Sie in Ihrem Homeverzeichnis einfach das Unterverzeichnis `bilder` mit `mkdir bilder` an. Dann lassen Sie sich mit `ls -l` den Inhalt anzeigen. Die Ausgabezeile für das Verzeichnis sieht dann etwa so aus:

```
drwxr-xr-x    2 hansi  benutzer      35 Jul 16 16:00 bilder/
```

Betrachten wir nun die Buchstabenfolge am Anfang. Der erste Buchstabe besagt, daß es sich um ein Verzeichnis handelt (`d` = directory). Dann folgen die entscheidenden neun Buchstaben, die die Rechte anzeigen. Die ersten drei stehen für den Besitzer, die nächsten drei für die Gruppe und die letzten drei für den Rest der Welt. `r` bedeutet lesbar, `w` beschreibbar und `x` heißt ausführbar. Ausführbar müssen Programmdateien und Verzeichnisse sein. Die Folge `rwxr-xr-x` heißt also, daß `ijhansij/ij` als Besitzer das Verzeichnis lesen, beschreiben und ausführen darf. Alle anderen (die Gruppe und der Rest) dürfen lesen und ausführen. Ausführen bedeutet bei Verzeichnissen, daß man in sie wechseln darf. Das Setzen der Rechte geschieht auf zwei verschiedene Weisen: Mit Symbolen für die verschiedenen Benutzer und die verschiedenen Rechte oder mit einer sogenannten Oktalzahl.

Symbole: Für User, Gruppe und Alle stehen die Symbole `u`, `g` und `a`, für die Rechte `r` (read), `w` (write) und `x` (executable). Setzen und Löschen von Rechten

geschieht mit Plus (+) und Minus (-). Soll die Datei `hans.pl` für alle lesbar werden, lautet das Kommando:

```
chmod a+r hans.pl
```

Zu beachten ist, daß das Setzen von Rechten für alle (a) das jeweilige Attribut sowohl für a, als auch u und g setzt, also an allen drei Positionen, das Setzen von Rechten für u oder für g aber jeweils nur an der ersten bzw. zweiten Stelle das Attribut setzt. Beim Setzen von Rechten für a kann im Befehl das a auch weggelassen werden. Es reicht also:

```
chmod +r hans.pl
```

Oktalzahl: Dieses Verfahren ist etwas eleganter, weil man alle Rechte für alle drei Gruppen gleichzeitig setzen kann. Man benutzt `chmod` mit einer dreistelligen Zahl, von der die erste Stelle für den User (u) steht, die zweite für die Gruppe (g) und die dritte für alle (a). Das Attribut lesen hat die Zahl 4, schreiben die Zahl 2 und ausführen die Zahl 1. Beim Setzen mehrerer Rechte werden die jeweiligen Zahlen an der jeweiligen Position addiert. Als Beispiel wollen wir für die Datei `hans.pl` folgende Rechte setzen: Der User soll lesen, schreiben und ausführen dürfen, die Gruppe soll lesen und ausführen dürfen und der Rest soll nur lesen dürfen. Beim User werden also 4, 2 und 1 addiert, bei der Gruppe die 4 (für lesen) und die 1 (für ausführen) und der Rest erhält lediglich die 4 für lesen. Der Befehl lautet demnach:

```
chmod 754 hans.pl
```

7.4 Prozesse beenden

7.4.1 killall

Sie kennen das Problem von **Windows**: Ein Programm bleibt hängen, es stürzt ab. Das passiert zwar relativ selten in **Linux**, aber es kommt vor. Vor allem Beta-Versionen, die noch nicht völlig fehlerbereinigt sind, neigen häufig dazu. Was in **Linux** allerdings fast nie passiert, ist, daß Sie den Rechner neu booten müssen. Sie haben die Möglichkeit, das hängende Programm zu beenden (abzuschießen). Dazu existieren die Befehle `killall` und `kill`. Arbeitet man auf einer X-**Windows**-Oberfläche, so sind im Normalfall alle Programme außer dem abgestürzten startbar. Das bedeutet, man öffnet eine Konsole und kann in dieser das hängende Programm beenden. Hat sich z.B. **Netscape** aufgehängt, so wird es (und alle anderen Programme, die das Wort `netscape` im Namen haben) durch den Befehl

```
killall netscape
```

beendet. Nach einem solchen unsanften Beenden eines Prozesses existiert oft ein sog. Lockfile. Dies steht im Beispiel von Netscape im Verzeichnis `/.netscape` und heißt schlicht `lock`. Falls ein Programm nach dem Abschießen nämlich nicht mehr gestartet werden kann (kommt bei Netscape des öfteren vor), muß zuerst dieses Lockfile gelöscht werden. Falls `killall` keine Auswirkung zeigt, kann die Option `-9` benutzt werden, die noch unsanfter ist als der einfache Befehl.

7.4.2 kill

`kill` funktioniert wie `killall`, nur daß das Programm nicht mit seinem Namen eingegeben wird, sondern mit seiner Prozeßnummer, der sog. PID. Jedes Programm, jeder Befehl, der gestartet wird, sei es beim Booten oder durch den User, erhält eine individuelle Prozeßnummer. Die laufenden Prozesse werden mit dem Befehl `ps` angezeigt. Um alle Prozesse, die gestartet wurden, anzuzeigen, wird `ps -ax` ausgeführt. Will man nun wissen, welche Prozeßnummer z.B. Netscape hat, kann man nach die Ausgabe aller Prozesse durchscrollen oder elegant per `grep`-Befehl den Netscape-Prozeß herausfiltern. Der gesamte Befehl lautet dann:

```
ps auxwww | grep netscape
```

Je mehr `ws` eingegeben werden, umso länger werden die Ausgabezeilen (nützlich bei sehr langen Pfaden, die sonst abgeschnitten werden. Die Ausgabe sieht dann etwa so aus:

```
hansi 779 0.0 5.1 22284 1120 ? S 11:58 0:01 /opt/netscape/netscape
hansi 968 0.0 0.2 1676 78 pts/0 S 12:53 0:00 grep netscape
```

Zuerst steht der Benutzer, der den Prozeß gestartet hat, danach folgt schon die Prozeßnummer (PID). Wir sehen hier zwei Prozesse, da `grep` nach der Zeichenfolge `netscape` gesucht hat und der `grep`-Befehl selbst ebenfalls ein Prozeß ist, der eine Nummer erhalten hat. Zum Beenden von Netscape verwenden wir nun den Befehl `kill` mit der PID des Prozesses, also in unserem Beispiel:

```
kill 11779
```

Auch hier kann bei hartnäckigen Fällen die Option `-9` benutzt werden, also `kill -9 11779`.

7.5 Automatische Namensergänzung und History

Zum Schluß betrachten wir noch zwei Funktionen, die einem die Arbeit auf der Shell (Befehlszeile) erleichtern. Die Funktion der Befehlshistory wird der eine oder

andere noch aus DOS kennen. Dort mußte die Funktion `doskey` aktiviert werden. Dann erhielt man per Cursortasten die Befehle, die vorher eingegeben wurden. In Linux ist diese Funktion in den verschiedenen Shells (in Linux existieren mehrere Befehlszeilenprogramme, sog. Shells, z.B. `bash` oder `sh`) implementiert. Drücken Sie einfach die Cursor-nach-oben-Taste und Sie werden feststellen, daß bei jedem Druck ein anderer Befehl erscheint. Diese History speichert standardmäßig bei SuSE Linux die 500 letzten Befehle.

Eine weitere Arbeitserleichterung bietet die Befehls- und Pfadergänzung. Die Shell kennt alle installierten Programme. Ebenso kennt sie alle Verzeichnisse des Rechners und damit alle Pfade. Daher ist es möglich, die ersten Buchstaben von Befehlen und von Verzeichnissen einzugeben und dann mit der Tabulatortaste die Shell aufzufordern, den Befehl oder den Verzeichnisnamen zu vervollständigen. Gibt es mehrere Möglichkeiten, ergänzt die Shell bis vor das Zeichen, das nicht mehr eindeutig ist. Besonders hilfreich ist diese Funktion bei sehr langen Pfaden, indem man immer nur die ersten zwei bis drei Zeichen des nächsten Unterverzeichnisses eingibt und die Shell den Rest ergänzen läßt. Auch bei Programmen mit Namen, die man sich schlecht merken kann, kann sich die Funktion auszahlen. Hat man z.B. den Mpegplayer `mpeg_play` installiert, ist sich aber in der Schreibweise nicht sicher (ob `mpeg_play`, `mpegplay`, `mpeg`, `mpegplayer`...), reicht es, `mpe` einzugeben und die Tab-Taste zu drücken und schon erscheint der Befehl in der richtigen Schreibweise.

Kapitel 8

Windowmanager

Autorin: *Jana Jaeger*

Wer von einem Linux System redet, meint ein Gesamtsystem aus vielen interagierenden Bestandteilen. Insbesondere die Softwarekomponenten sind weitgehend voneinander unabhgig zu betreiben, was dem Benutzer ein ungewöhnlich hohes Maß an Freiheit bei der Auswahl der einzelnen Anwendungen läßt. Die Vielfalt der unter Linux erhältlichen Windowmanager ist nur ein Beleg dafür.

8.1 Was tut ein Windowmanager?

Bevor der Benutzer komfortabel zwischen einzelnen Fenstern, virtuellen Desktops und Anwendungen hin- und herklicken kann, müssen erst einmal die einzelnen Schichten seines Systems perfekt miteinander kommunizieren können. Auf unterster Ebene findet sich die Hardware (in diesem Fall die Grafikkarte), die über den X-Server mit der Softwareebene in Kontakt steht. Der X-Server als solcher hilft noch nicht sehr viel weiter. Erst im Zusammenspiel mit sogenannten X-Clients (den eigentlichen Anwendungsprogrammen), die die im X-Server eingebauten Routinen zur Darstellung von Grafiken und grafischen Oberflächen nutzen können, ist Kommunikation zwischen der Hardware- und der Software möglich. Der X-Server und der/die X-Client/s kommunizieren über das Netz miteinander, d.h. es ist möglich, den X-Client auf einem anderen Rechner zu starten und seine Bildschirmausgabe auf einem simplen Terminal anzuzeigen. Wirklich komfortables Arbeiten ist auch mit bloßen X-Clients noch nicht möglich. Hier kommt der Windowmanager ins Spiel. Einen groben Überblick über seine Aufgaben gibt die folgende Auflistung:

- Aussehen der Fenster
 - Breite, Farbe und 3D-Effekte des Rahmens

- Bedienelemente zum Verschieben oder Vergrößern des Fensters, Titelseite und Schriftarten
- Überlagern von Fenstern
 - Raising Verhalten, z.B. Autoraising (Hervortreten in den Vordergrund)
 - Anpinnen von Fenstern
- Fokussieren eines Fensters durch
 - Anklicken
 - "Berührung" mit dem Mauszeiger
- PopUpmenüs
 - Aussehen der Menüs (Farbe, Schriftart)
 - Verhalten der Menüs und Untermenüs
- Bildschirmhintergrund
- Virtueller Desktop (mehrere Desktops/Screens)
- Verwaltung von Icons
- Verbindung von Sounds mit Ereignissen auf dem Desktop

8.2 Kurzportraits

Die folgende Sammlung einiger Kurzportraits der bekanntesten Windowmanager erhebt keinerlei Anspruch auf Vollständigkeit. Eine Liste mit den jeweils aktuellen Versionen der meisten Windowmanager findet sich bei Freshmeat. Die Kurzportraits sollen einzig und allein einen kleinen Einblick in die Fülle der verfügbaren Programme geben und zum "Selberausprobieren" animieren. Besonders wichtig ist dabei, die Ängste vor kryptischen Konfigurationsdateien abzubauen. Hier helfen im Web erhältliche Beispielfiguren (Links werden unten angegeben), die seltsam anmutenden Zeilen zu verstehen und nachher auch selbst abzuwandeln. Einige wenige Windowmanager (*WindowMaker* und *Enlightenment*) bieten dem konfigurationswütigen Benutzer ein grafisches, "intuitives" Konfigurationsprogramm.

8.2.1 Ctwm

Ctwm ist ein relativ schnörkelloser, aber dafür sehr funktionaler Windowmanager, der als Erweiterung eines der "Dinosaurier" unter den Windowmanagern, Twm, entstanden ist. Er unterstützt das Arbeiten mit bis zu 32 virtuellen Desktops ("workspaces"), die alle unterschiedlich konfiguriert sein können (Farben, Namen, Aussehen der Buttons und root-Shells). Der Wechsel zwischen einzelnen Desktops geschieht entweder über einen Funktionsaufruf oder über einen optionalen "Workspace Manager". Titel und Fensterränder gibt es auf Wunsch auch in 3D-Darstellung. Buttons, Icons und die Hintergründe des Rootfensters sind animierbar. Menüs lassen sich am Desktop festpinnen. Fenstertitel werden optional in voller Breite, als eine Art Aktenreiter oder gar nicht angezeigt. Genauere Informationen über Ctwm gibt die offizielle Ctwm Website. In begrenztem Maß können hier auch Themes für Ctwm heruntergeladen werden. Beispielkonfigurationen samt Screenshots können über <http://www.plig.org/xwinman/ctwm.html> bezogen werden.

8.2.2 Fvwm/Fvwm95

Über lange Zeit hinweg war Fvwm der dominierende Windowmanager auf Linux Systemen. Bei niedrigem Speicherbedarf bietet fvwm trotzdem eine einfache 3D-Oberfläche und virtuelle Desktops. Der Grundaufbau ist durch verschiedene Module erweiterbar. Wahlweise läßt sich Fvwm per Maus oder Tastatur bedienen. Autoraising beherrscht Fvwm ebenso wie Fokussieren von Fenstern und Desktopwechsel beim Herausbewegen des Mausursors aus dem Bereich des aktuellen Desktops. Detaillierte Informationen zu Fvwm gibt die Homepage des Fvwm-Projekts (<http://www.fvwm.org>). Hier liegen die jeweils stabilen Versionen bereit zum Download. Verschiedene Dekorationsstile für Fensterleisten und Sounds können hier auch heruntergeladen werden. Unter <http://www.plig.org/xwinman/fvwm.html> sind Screenshots sowie Beispielkonfigurationsdateien sowohl für Fvwm als auch für Fvwm2 abzurufen. Die .fvwmrc wird viele Anwender auf den ersten Blick durch ihre kryptischen Aufbau verschrecken.

Fvwm95 ist eine Art *Hack* von Fvwm2, der das Aussehen eines Windows-Desktops emuliert. So können Umsteiger sanfter in eine *fremde* Welt hinübergleiten. Näheres findet sich auf der offiziellen Homepage von Fvwm95: (<ftp://mitac11.uia.ac.be/html-test/fvwm95.html>).

8.2.3 AfterStep

AfterStep baut auf Fvwm auf, emuliert aber das Aussehen und den Bedienkomfort der NEXTSTEP© Benutzeroberfläche. Besonderes Bonbon von AfterStep ist seine Buttonleiste "Wharf", die mit vielen verschiedenen Applikationen ausgebaut

werden kann (zum Beispiel mit Applikationen zur Systemüberwachung). Da *AfterStep* auf *Fvwm* aufbaut, sind noch viele Module aus *Fvwm1.x*-Zeiten auch unter *AfterStep* verwendbar. Insgesamt macht *AfterStep* einen sehr bunten, aber auch funktionalen und robusten Eindruck. Ausgewählte Screenshots mit den entsprechenden Konfigurationsdateien sind über

<http://www.plig.org/xwinman/afterstep.html>

erhältlich. Umfassende Informationen zu *AfterStep*, noch mehr Screenshots, neue Icons und downloadbare Programme finden sich auf der offiziellen Homepage des *AfterStep* Projekts (<http://www.afterstep.org>). Themes sind hier zu finden:

<http://as.themes.org>

8.2.4 Enlightenment

Enlightenment basiert ursprünglich auf *Fvwm2*, mittlerweile ist aber davon nicht mehr viel zu sehen. *Enlightenment* ist gemessen an seinem Bedienungskomfort und seinem optischen Erscheinungsbild ein schlankes und schnelles Programm. Es bietet dem Benutzer eine Fülle von Möglichkeiten, den Desktop nach eigenen Vorlieben umzugestalten (z.B. mit Themes, <http://e.themes.org>). *Enlightenment* steckt voller grafischer Spielereien.

Auf <http://www.enlightenment.org> findet sich alles Wissenswerte zum Thema *Enlightenment*. *Gnome* unterstützt *Enlightenment* in vollem Umfang, auch *KDE* wird unterstützt. Zukünftig ist sogar an einen eigenen Filemanager gedacht, womit die Grenze vom Windowmanager zum Desktop verschwimmt.

8.2.5 IceWM

IceWM ist extrem flexibel und schnell. Dieser Windowmanager kann auch ohne Mausunterstützung arbeiten. *IceWM* unterstützt *Gnome*, kann aber auch ohne *Gnome* eine Reihe sehr komfortabler Bedienelemente vorweisen. Zentral hierbei die vielseitige Taskleiste am unteren Bildschirmrand. Von dort läßt sich zwischen den virtuellen Desktops wechseln, eine Anzeige informiert über die Art und Anzahl der offenen Fenster und laufenden Applikationen, es lassen sich Applikationen mit ihren Icons in die Taskleiste legen und vieles andere mehr. *IceWM* unterstützt Themes (<http://icewm.themes.org>). Weiterführende "Literatur" für *IceWM* ist unter <http://www.icewm.org> zu finden.

8.2.6 WindowMaker

Seiner Optik nach ist *WindowMaker* an *AfterStep* angelehnt. Der Bedienungskomfort entspricht demjenigen von NEXTSTEP©. *WindowMaker* fällt auch durch seine Schnelligkeit und seinen geringen Speicherhunger auf. Die Zusammenarbeit von *Gnome* und *KDE* Anwendungen mit *WindowMaker* funktioniert sehr gut. Wie schon für *AfterStep* mit "Wharf" hat auch *WindowMaker* ein Dock

zum Anbinden von Applikationen. Dieses unterstützt sogar "Drag-and-Drop". Das Konfigurieren des Windowmanagers erfolgt über eine grafische Benutzeroberfläche. Beispielkonfigurationen und die daraus resultierenden Screenshots sind unter <http://www.plig.org/xwinman/wmaker.html> zu finden. Themes und weitere Informationen bieten die offizielle Homepage, <http://www.windowmaker.org/>, und <http://wm.themes.org>.

8.2.7 XFce

XFce ist ein extrem schlanker Desktop für alle möglichen UNIX-Systeme. Er enthält einen eigenen Windowmanager, ein Panel, einen eigenen Filemanager und viele andere nützliche kleine Applikationen. Darunter auch ein Modul zur Eingliederung von **Gnome** (Anwendungen). Die wichtigsten Informationen zu *XFce* finden sich unter <http://www.xfce.org>.

8.2.8 Gnome

Gnome ist eine freie Desktopumgebung für alle Arten von UNIX-artigen Betriebssystemen. **Gnome** läßt dem Anwender die Wahl, welcher Windowmanager verwendet werden soll. Von Hause aus besitzt **Gnome** keinen eigenen Windowmanager, kann aber mit den meisten der genannten Beispiele kombiniert werden.

8.2.9 KDE

KDE ist eine freie Benutzeroberfläche mit integriertem Windowmanager. Eine Fülle von KDE relevanten Informationen befindet sich auf den Webseiten des KDE-Projekts (<http://www.kde.org>).

8.3 Wo hört ein Windowmanager auf und wo beginnt ein Desktop?

Wie schon oben bei den Kurzportraits erkennbar, ist die Grenze zwischen einem gut ausgestatteten Windowmanager und einem minimalen Desktop fließend. Ein paar Unterschiede lassen sich aber bei genauerem Hinsehen doch noch ausmachen.

Windowmanager müssen im Grunde alle Funktionen enthalten, die es ermöglichen, Fenster auf dem Bildschirm hin- und herzubewegen, zwischen einzelnen einzelnen Bildschirmen (oder Desktop) zu wechseln und Anwendungen parallel und unabhängig voneinander auf einem oder mehreren Desktops auszugeben, praktische Menüs und Icons zu benutzen und vieles andere mehr. Desktops gehen darüber hinaus, indem sie Interaktionen zwischen einzelnen Anwendungen unterstützen. Das wäre zum Beispiel "Drag-and-Drop", "Copy-and-Paste" und

die Verknüpfung von Applikationen. KDE und Gnome bieten eigene Filemanager (bald auch *Enlightenment*) und Sessionmanagement.

Der Desktop ist für alles zuständig, was im Unsichtbaren das Leben des Benutzers erleichtert. Das heißt, diese Funktionen (insbesondere das Sessionmanagement) sieht der Benutzer nicht bei der Arbeit, würde sie aber schmerzlich vermissen, wenn es sie nicht gäbe. Der Windowmanager ist für alles das zuständig, was im sichtbaren Bereich die Benutzerfreundlichkeit eines Betriebssystems erhöht. Wirklich sinnvoll ist die Benutzung eines Desktops dementsprechend nur in Kombination mit einem Windowmanager. Das Zusammenspiel zwischen Desktop und Windowmanager ähnelt stark einer Symbiose - keiner der beiden ist ohne den anderen wirklich sinnvoll einzusetzen, will man maximalen Komfort. GNOME enthält keinen eigenen Windowmanager, unterstützt aber viele der oben genannten Beispiele. KDE liefert seinen eigenen Windowmanager *kwm* zwar mit, läßt sich aber (begrenzt) auch mit beliebigen anderen Windowmanagern zusammen einsetzen.

Kapitel 9

With a little help from my friends ...

Autorin: *Jana Jaeger*

So, jetzt ist es passiert. Eigentlich sollte das ein Moment zum Feiern sein - Linux auf dem eigenen Rechner, selbstinstalliert - naja, eigentlich, denn es will nicht so, wie es sollte. Vielleicht hüllt der Rechner sich jetzt in vornehmes Schweigen, wo er doch in Vor-Linux-Zeiten Sound in Hülle und Fülle von sich gegeben hatte. Oder er mag plötzlich nichts mehr von der restlichen Welt wissen und verzichtet auf so prosaische Dinge wie seine Netzwerkverbindung oder ...

Solche und tausend kleine andere Ärgernisse können die Entdeckerfreude am heimischen Linux-Rechner ganz schön verhageln. Was nun? Wo gibt es Hilfe?

9.1 Hilfe zur Selbsthilfe?

Zuallererst liegt da ein Stapel Dokumentation, vielleicht sollte man eher sagen, fast zwei Kilo davon (das ist die Gewichtsangabe für SuSE Linux, für andere Produkte gilt Entsprechendes). Eventuell ist das Problem schon bekannt und dokumentiert. Wem die Bücher zu schwer sind, der kann sich Doku - vorausgesetzt, die Installation hat soweit geklappt, daß ein Terminalfenster oder noch besser ein Browser (z.B. *Konqueror*) startbar ist, am Rechner nachlesen und komfortabel durchsuchen. Die Quellen für das SuSE Handbuch z.B. sind unter `/usr/share/doc/suse/suselxde` aufzurufen. Die Dokumentation für einzelne Programmpakete findet sich unter `/usr/share/doc/packages/Paketname`.

Findet sich hier nichts, gibt es noch die Möglichkeit, in den vielen Kurzanleitungen des Linux Documentation Projects, den HOWTOs, zu stöbern. Viele der wichtigsten HOWTOs gibt es mittlerweile auch in Deutsch, und eine Auswahl sowohl der deutschen als auch der englischsprachigen HOWTOs sind im Umfang der Distribution enthalten (`/usr/share/doc/howto/`). Der aktuellen Bestand ist

jederzeit unter <http://www.linuxdoc.org> abzurufen. Ein Überblick über die zur Zeit verfügbaren deutschen HOWTOS und das Deutsche HOWTO Projekt sind jederzeit über die Seiten des Deutschen Linux HOWTO Projekts (<http://www.tu-harburg.de/dlhp>) zu beziehen.

Die rudimentärsten Informationen in komprimierter Form zu (fast) allen Befehlen lassen sich über die Man- oder Infopages dieser Befehle abrufen. Mit `Konqueror` reicht die einfache Eingabe von `man:befehl` bzw. `info:befehl` und schon stellt er die Seiten in optisch angenehm aufbereiteter Form dar. Natürlich lassen sich diese Seiten auch von der Kommandozeile aus starten (mit `man Befehl` bzw. `info Befehl`). Für den Anfang sind sie meist ein wenig zu knapp gehalten, leisten aber nachher als Nachschlagewerke im täglichen Betrieb wertvolle Dienste, wenn man mal eben eine Aufrufoption nachgucken möchte. Für den Anfänger kranken die Man- oder Info-Pages meist massiv daran, daß er Anwendungsbeispiele aus dem Einsatz der verschiedenen Programme vermißt.

Benutzer von **SuSE Linux** können viele schon häufig im Rahmen des **SuSE-Supports** aufgetretenen Fragen und Probleme offline in der Supportdatenbank über das **SuSE Hilfesystem** nutzen. Der aktuelle Stand läßt sich online über <http://sdb.suse.de/de/sdb/html/index.html> abrufen.

9.2 Hardware-Informationen

Vor dem Kauf neuer Hardware oder bei begründetem Verdacht darauf, daß die eigene Hardware vielleicht nicht von **Linux** unterstützt sein könnte, gibt es diverse Seiten im Internet, die Hardware Infos bündeln. Nicht zuletzt unterhält **SuSE** eine eigene Datenbank (<http://cdb.suse.de>) mit einzelnen Komponenten und deren Unterstützungstatus, soweit er den Supportmitarbeitern und Entwicklern bekannt ist. Eine kleine Auswahl hardwarerelevanter Links (Danke an Stephan Barth vom **SuSE** Installationssupport, der diese Liste bereitgestellt hat):

• Hardware allgemein

- <http://www.linux-usb.org/>
The Linux USB Project. Die USB-Seite für Linux
- <http://www.qbik.ch/usb/devices/>
Unterstützte USB-Geräte. Eine ständig aktualisierte Liste unterstützter Hardware.
- `/usr/src/linux/Documentation/`
Treiberdokumentation. Die lokale Dokumentation zu den Treibern, die mit dem Kernel mitgeliefert werden. Dafür müssen auf Ihrem Rechner die Kernelquellen (`lx_suse` aus der Serie `d`) installiert sein.
- `/usr/src/linux/drivers`
Auch die Kerneltreibersourcen (lokal) enthalten nützliche Hinweise

und Dokumentation. Hierfür müssen ebenfalls die Kernelquellen installiert sein.

- <http://www.heise.de/ct/treiber/>
c't Treiber Service. Hier findet man jede Menge Links zu Hardware- und Softwareherstellern.
- <http://members.hyperlink.net.au/chart/download/vendors.txt>
Vendor IDs, die dem BIOS gemeldet werden. Falls der Kernel ein Gerät einmal nicht identifizieren kann, kann man hier nach dem Hersteller schauen. Die IDs sieht man z.B. bei `lspci -vv`.
- <http://www.hwb.acc.umu.se>
The Hardware Book. Hier findet man u.a. Steckerbelegungen (PCs, Homecomputer, Video etc.), technische Info zu Bussystemen (ISA, EISA, PCI-Varianten, Nubus, Zorro, etc.), Anleitungen zum Basteln von Adapterkabeln, aber auch Schaltpläne für aktive Audiofilter.
- <http://www.in-berlin.de/user/scorpio/faqkabel.html>
Deutsche Kabel-FAQ. Diverse Steckerbelegungen, Kabeldaten, Signalbezeichnungen, -spannungen und -stromstärken, sowie spezielle Schaltungen.

• IDE / ATAPI

- <http://ata-atapi.com/>
Allgemeines zum Thema ATAPI und IDE. Sehr gut!

• Sound

- </usr/src/linux/Documentation/sound/>
Die sehr gute Dokumentation (lokal) zu den Kerneltreibern für Soundkarten. Dafür müssen auf Ihrem Rechner die Kernelquellen (1x_suse aus der Serie d) installiert sein.
- <http://www.alsa-project.org/>
Advanced Linux Sound Architecture (ALSA). Das ALSA-Projekt, das mittlerweile der Standard für die Soundunterstützung ist.

• Scanner

- <http://www.mostang.com/sane/>
SANE Home Page. Das Projekt, das praktisch alle Scannertreiber schreibt. Das heißt, was von SANE nicht unterstützt wird, wird von Linux nicht unterstützt.
- <http://www.buzzard.org.uk/jonathan/scanners.html>
Parallelport Scanner unter Linux

- <http://www.buzzard.org.uk/jonathan/scanners-usb.html>
Die leider noch sehr kurze Liste der unterstützten USB-Scanner

- **Netzwerkkarten**

- </usr/share/doc/howto/en/html/Ethernet-HOWTO.html>
Das Ethernet-HOWTO (lokal). Ausführliche Hinweise auch zur Konfiguration von mehreren Netzwerkkarten.
- </usr/src/linux/Documentation/networking/>
Die Dokumentation (lokal) zu den Kernaltreibern für Netzwerkkarten. Dafür müssen auf Ihrem Rechner die Kernelquellen (1x.suse aus der Serie d) installiert sein.
- <http://www.scyld.com/network/>
Homepage des Netzwerkkartentreiberentwicklers Donald Becker, der Netzwerkkartenguru schlechthin. Die Dokumentation sieht auf den ersten Blick spärlich aus, der Schein trügt aber ;-)
- <http://www.linuxtr.net/>
Linux Token Ring Project. Es werden nur ein paar Token Ring Netzwerkkarten unterstützt, aber hier findet man eventuelle Patches. Weiter hilft hier auch das Token-Ring-HOWTO:
<file:/usr/share/doc/howto/en/html/mini/Token-Ring.html>
- <http://www.dlink.de/linux/index.asp>
Support-Seiten der D-Link Deutschland GmbH

- **Notebooks**

- </usr/share/doc/packages/pcmcia/>
Hier findet man sehr viel zu den unterstützten Karten und das PCMCIA-HOWTO.
- <http://www.linux-laptop.net/>
Linux on Laptops. Viele Links zu allgemeinen Fragen und zur Installation von bestimmten Notebooktypen.

- **RAID**

- </usr/share/doc/howto/en/html/Root-RAID-HOWTO.html>
Das HOWTO zu Software RAID
- </usr/share/doc/howto/en/html/Multi-Disk-HOWTO.html>
Das Multi-Disk-HOWTO geht auch viel und oft auf RAID ein.
- http://www.icp-vortex.com/supp/down/linux/linux_d.html
ICP Vortex. Diese Firma schreibt die Treiber zu ihren RAID-Controllern selbst. Hier findet man die neuesten Treiber.

- **Drucker**

- `/usr/share/doc/packages/ghostscript/catalog.devices`
Eine Auflistung aller von Ghostscript unterstützten Drucker
- `http://www.linuxprinting.org/printer_list.cgi`
Die Druckerliste schlechthin.
- `http://www.torque.net/parport/`
Linux Parallel Port Home Page

- **Modem**

- `http://www.idir.net/gromitkc/winmodem.html`
Die Modem-Seite schlechthin. Ist es ein Winmodem oder ein normales Modem, so daß man es unter Linux verwenden kann? Für die Liste an sich muß man auf BIG LIST auf der linken Seite klicken.

9.3 Hilfe vom Rest der Welt

9.3.1 Newsgroups

Die traditionellen (deutschsprachigen) Newsgroups zum Thema Unix/Linux sind die `de.comp.os.unix.linux` Gruppen. Eine FAQ zu dieser Hierarchie liegt als HTML-Version unter `http://www.tnt-computer.de/yanip/dcoul-faq/`. Fragen an die `de.comp.os.unix.linux`-Newsgroups sollte man allerdings erst stellen, nachdem man die Informationsquellen, die im eigenen System verfügbar sind, voll ausgeschöpft hat. Sonst kann es durchaus passieren, daß man eine Antwort vom Format **RTFM** (Read the f***ing manual) erhält. Außerdem liefern einschlägige Suchmaschinen oder das Deja Usenet Archiv unter `http://groups.google.com/groups?hl=en&lr=&safe=off&group=de` schon einen Überblick darüber, welche Fragen zum eigenen Problem schon gestellt wurden und darüber, ob die Lösung nicht womöglich schon einmal gepostet wurde.

Die Hierarchie der deutschen `comp.os.*` Gruppen im Überblick:

<code>de.comp.os.unix.networking.misc</code>	Netzwerke und Kommunikation
<code>de.comp.os.unix.networking.samba</code>	Unix und Windows tanzen - Samba
<code>de.comp.os.unix.x11</code>	Rund um das X Window System unter Unix
<code>de.comp.os.unix.shell</code>	Shell, Skripte, Tools
<code>de.comp.os.unix.apps.misc</code>	Allgemeines zu Anwendungen unter Unix
<code>de.comp.os.unix.apps.gnome</code>	GNOME - GNU Network Object Model Environment
<code>de.comp.os.unix.apps.kde</code>	KDE - K Desktop Environment
<code>de.comp.os.unix.programming</code>	Programmieren unter Unix ...
<code>de.comp.os.unix.discussion</code>	Fragen und Diskussionen, die in anderen Sparten nicht so gern gesehen werden (Glaubenskriege, etc.)
<code>de.comp.os.unix.misc</code>	Unixthemen, die in keine der oben genannten Gruppen passen, aber keine eigene Gruppe haben

Der Linux-Ast in dieser Hierarchie verzweigt sich selbst dann wieder:

<code>de.comp.os.unix.linux.hardware</code>	Hardwarefragen zu Linux
<code>de.comp.os.unix.linux.moderated</code>	Moderierte Gruppe zu Linux
<code>de.comp.os.unix.linux.misc</code>	s.o., alles, was sonst nirgends unterkommt
<code>de.comp.os.unix.linux.infos</code>	Informationen zu Linux

Einen sehr guten Überblick über alle möglichen Newsgroups im deutschsprachigen Usenet mit Namen, Kurzbeschreibung, Kommentar (und wenn verfügbar der Charta) der jeweiligen Gruppe bieten die *dana* (de.admin.news.announce) Seiten (<http://www.dana.de/mod/chartas/de.comp.html>).

Bei Postings in den oben beschriebenen Gruppen sollte man immer darauf achten, jeweils die Gruppe auszuwählen, die das Thema am besten eingrenzt - das Posten der gleichen Frage in mehreren Gruppen gleichzeitig (Crossposten) ist absolut verpönt.

Probleme und Fragen zu einzelnen Anwendungen gehören eher nicht in die `de.comp.os.linux.*` Ecke, sondern eher in Gruppen, die sich extra dafür vorgesehen sind.

Netzwerkfragen sind nicht zwingend linuxspezifisch, deshalb sollten sie eher nicht in einer der `de.comp.os.unix.linux*` Newsgroups erscheinen, sondern in `de.comp.os.unix.networking.misc` (Beispiele: NFS, PPP, etc.). Samba wird sogar in einer eigenen Untergruppe thematisiert.

Ein kleiner Überblick über kommunikationstechnisch relevante Newsgroups:

<code>de.comp.os.unix.networking.misc</code>	Netzwerke und Kommunikation
<code>de.comp.unix.networking.samba</code>	Unix und Windows tanzen - Samba
<code>de.alt.comm.isdn4linux</code>	zum Thema ISDN unter Linux
<code>de.alt.comm.mgetty</code>	für die Faxsoftware <i>mgetty</i> und <i>sendfax</i>
<code>de.comm.software.mailreader.misc</code>	für Mailreader (Pine und Co.)
<code>de.comm.software.mailserver</code>	für Mailserver (Sendmail, etc.)
<code>de.comm.software.newsreader</code>	für Newsreader (slrn, ...)
<code>de.comm.software.newsserver</code>	für newsserver (leafnode, ...)
<code>de.comm.infosystems.www.browsers</code>	für WWW-Clients (Netscape, etc.)
<code>de.comm.infosystems.www.servers</code>	für WWW-Server (Apache und Co.)
<code>de.comm.infosystems.misc</code>	für Infosysteme (wie Gopher, etc.)
<code>de.comp.security.misc</code>	Computer- und Datensicherheit
<code>de.comp.security.firewall</code>	Sicherheit trotz Netz

9.3.2 SuSE-Mailinglisten

SuSE unterhält zahlreiche, für jeden Interessenten offene Mailinglisten zu verschiedenen Themenschwerpunkten:

<code>suse-linux</code>	Diskussionen rund um SuSE Linux (auf Deutsch)
<code>suse-security</code>	Security Diskussionen (allerdings in englischer Sprache)
<code>suse-security-announce</code>	Hier werden die SuSE Security-Updates veröffentlicht (englisch)
<code>suse-isdn</code>	ISDN mit SuSE Linux
<code>suse-laptop</code>	SuSE Linux auf Laptops
<code>suse-announce</code>	Ankündigungen und Infos von SuSE

Einen Überblick erreicht man unter:

<http://www.suse.de/de/support/maillinglists/index.html>

Diese Seite gibt Ihnen auch die Möglichkeit, sich direkt für die verschiedensten Mailinglisten anzumelden.

9.3.3 (Installations-) Support

Hilft weder der tiefere Blick in die entsprechende Dokumentation, noch der Austausch mit Gleichgesinnten, bleibt oft nur noch die Nutzung eines von vielen Linux-Unternehmen angebotenen Support-Services. Wie und in welchem Umfang Fragen beantwortet werden und was die Bearbeitung im einzelnen für Kosten mit sich bringt, hängt stark vom Anbieter ab.

SuSE Kunden haben nach Kauf einer neuen Distribution Anspruch auf kostenlosen Installationssupport (für SuSE Linux Personal 60 und für SuSE Linux Professional 90 Tage). Alle Fragen, die sich um einen lauffähigen Heim-PC mit einfacher Internetanbindung drehen, werden vom Supportteam möglichst unge-

hend am Telefon oder per E-Mail beantwortet. Den genauen Umfang des Installationssupports und die Formalitäten, die zur Inanspruchnahme des I-Supports nötig sind, werden auf den SuSE-Webseiten beschrieben:

- <http://support.suse.de/de/services/faq-isupport.html#quickstart>,
Fragen und Antworten zum Installationssupport
- <http://support.suse.de/de/register/>,
Software-Registrierung
- <http://support.suse.de/de/services/anfrage-onlineform.html>,
SuSE Installationssupport-Anfrage

9.4 ... in eigener Sache ...

Natürlich gibt es auch zahlreiche Linux-Websites, wie zum Beispiel diese hier, die eine Fülle von Informationen vorhalten. Eine Übersicht findet sich hier:

<http://www.fokus.gmd.de/linux/linux-servers.html>

Leider haben solche Angebote meist (subjektiv) den Nachteil, daß sie sich gerade **nicht** mit der Thematik beschäftigen, die einem so brennend auf der Seele lastet. Meistens kommt Hilfe in solchen Fällen dann eher von einem Wesen aus Fleisch und Blut als irgendwo aus dem Netz. Sei es ein hilfsbereiter Crack, der den Hilferuf in einer Newsgroup oder auf einer Mailingliste beantwortet, ein Bekannter, der mit seinem Wissen aushelfen konnte, oder auch der Supportmitarbeiter, der am Telefon oder per Mail die rettenden Auskünfte gibt.

Alle Portale, Websites oder Foren im Internet, die sich im Internet mit Linux auseinandersetzen, leben davon, daß ihre Leser ihre Erfahrungen beisteuern (sei es in konkreten HOWTOs oder kurzen Tips, oder über eine Mail mit einem persönlichen Erfahrungsbericht). So werden die eigenen schmerzlichen Erfahrungen anderen erspart, die vielleicht wiederum Problemlösungen parat haben, auf die man selbst eines Tages gerne zurückgreifen möchte... Letztendlich bedeutet das für beide Seiten ein fröhliches Geben und Nehmen.

Ohne die Beiträge der einzelnen Benutzer und Tester wäre weder Linux selbst das geworden, was es mittlerweile ist, noch gäbe es Dokumentationsprojekte irgendeiner Art oder vernünftige Foren oder ähnliche Plattformen!

Literaturverzeichnis

- [1] Linux in a Nutshell, O'Reilly
- [2] Deaver, Jeffery: Lautloses Duell, Goldmann 2002
- [3] Tom DeMarco: Der Termin, Hanser Verlag 1998
- [4] Hafner, Katie; Lyon, Matthew: where wizards stay up late, Touchstone 1998
- [5] Kofler: Einführung in Linux, 1. Auflage, Hanser Verlag 1996
- [6] Kopka, Helmut: \LaTeX Einführung Band 1, 2.Auflage, Addison-Wesley 1996
- [7] Stoll, Clifford: Das Kuckucksei, Fischer Verlag 1989
- [8] Stoll, Clifford: The Cuckoo's Egg, Pocket Books 1989
- [9] Torvalds, Linus: Just for fun, Hanser Verlag 2001

Anhang A

GNU Free Documentation License

Version 1.2, November 2002

Copyright ©2000,2001,2002 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document *free* in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of *copyleft*, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The **Document**, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as **you**. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A **Modified Version** of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A **Secondary Section** is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The **Invariant Sections** are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The **Cover Texts** are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A **Transparent** copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not Transparent is called **Opaque**.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats

include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The **Title Page** means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, *Title Page* means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section **Entitled XYZ** means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as **Acknowledgements**, **Dedications**, **Endorsements**, or **History**.) To **Preserve the Title** of such a section when you modify the Document means that it remains a section *Entitled XYZ* according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly

identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled *History*, Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled *History* in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the *History* section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled *Acknowledgements* or *Dedications*, Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled *Endorsements*. Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled *Endorsements* or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document,

you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled *Endorsements*, provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled *History* in the various original documents, forming one section Entitled *History*; likewise combine any sections Entitled *Acknowledgements*, and any sections Entitled *Dedications*. You must delete all sections Entitled *Endorsements*.

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License

in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an *aggregate* if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled *Acknowledgements*, *Dedications*, or *History*, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License *or any later version* applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright ©YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled *GNU Free Documentation License*.

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the *with...Texts.* line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.