

KAPITEL 1

Einführung in Linux

Linux ist in aller Munde! *Linux* läuft auf fast jedem modernen Computer, sei es ein PDA (Personal Digital Assistant), ein Laptop, ein Bürocomputer oder ein Mainframe. *Linux* ist sowohl der Name für einen Kernel, als auch für ein Betriebssystem, als auch für eine vollständige Arbeitsumgebung. Es ist für den Endanwender mittlerweile schwierig zu entscheiden, ob *Linux* oder ein anderes Betriebssystem und dessen Programme auf einem Computer im Einsatz sind. Programme und Benutzeroberflächen verschiedenen Ursprungs haben sich im Funktionsumfang und Leistung so weit angenähert, dass allenfalls das Design einen Hinweis auf den Ursprung bietet.

Die Verwendung von *Linux* bietet dem Endanwender große Vorteile – einerseits kann alles so belassen werden, wie es der Lieferant voreingestellt hat, andererseits können Erscheinungsweise und Funktion aber auch vollständig geändert und den wirklichen Bedürfnissen des Benutzers angepasst werden. Es gibt viele Anbieter mit völlig unterschiedlichen Geschäftsmodellen, so dass der Benutzer das auswählen kann, was ihm am besten zupass kommt. Wer möchte, kann mit einem Internetzugang ohne jede Abhängigkeit von einem Lieferanten sich sein persönliches *Linux*-System selbst zusammenstellen.¹

Die Idee eines finnischen, schwedisch sprechenden Informatikstudenten im Jahre 1989 war es, so etwas wie einen dritten Weg in die erstarrten Welten der Softwareproduktion einzuführen – auf der einen Seite waren die Anbieter kommerzieller Software, deren Produkt- und Lizenzgestaltung sich nicht notwendigerweise an den Anwenderinteressen orientiert – auf der anderen Seite war die akademisch strenge, in sich geschlossene Welt der Hochschulen und Forschungsinstitute wie der University of Berkeley und des Massachusetts Institute of Technology.

Linus Torvalds revolutionärer Ansatz war es, auf Basis des Internets jedem das Mitmachen zu ermöglichen, ohne eine hohe finanziellen Eintrittshürde überwinden zu müssen, wie es bei kommerzieller Software der Fall ist oder zu akademischer Anerkennung gelangt zu sein, wie es bei den Projekten der akademischen Institutionen üblich ist. Eine gewisse Hemdsärmeligkeit hat sich *Linux* bewahren können, das macht auch einen Teil seines Charmes aus.

¹ <http://www.linuxfromscratch.org/>

Über dieses Buch

Mit diesem Buch wird der Versuch unternommen, den Stab von »*Linux* – Installation und Konfiguration« zu übernehmen und weiterzutragen. *Linux* hat sich wie seine Benutzer weiterentwickelt und Verlag und Autoren wollen den geänderten Anforderungen genügen.

An wen richtet sich dieses Buch

Dieses Buch soll alle Interessenten für *Linux* ansprechen, die ihren *Linux*-Computer verstehen und zielgerichtet einsetzen wollen.

Linux erlaubt es dem Interessierten, sich einen Entwurf von Betriebssystem und Software bis ins kleinste Detail anzuschauen oder auch nur damit spielerisch umzugehen. *Linux* verwirklicht sowohl klassische wie aktuelle Ansätze.

Linux bietet auch dem reinen Computerspieler eine leistungsfähige Plattform, die dem Spielen über das Netz eine interessante Plattform bietet.

Leseweise für dieses Buch

Die Kapitel sind nach Möglichkeit in sich geschlossen. Wenn verwendete Begriffe einmal nicht vertraut sind, bietet das Glossar im Anhang eine Hilfe. Natürlich sind Inhalte über die Installation einer Firewall ohne grundlegende Netzwerkkennntnisse nicht zu verstehen, so dass die weiter hinten liegenden Kapitel mit den eher fortgeschrittenen Themen erst dann dem Leser Nutzen bringen, wenn ihm die Kenntnisse aus den weiter vorne gelegenen Kapiteln zur Verfügung stehen.

Rolle dieses Buches in der O'Reilly-Bibliothek

Dieses Buch soll einen Einstieg in das Thema *Linux* erleichtern. Viele Bücher zu Spezialthemen sind im O'Reilly-Verlag verfügbar, die alle Themen dieses Buches jeweils einzeln in größerer Tiefe abhandeln.

Typographische Konventionen

Folgende typographische Konventionen werden in diesem Buch verwendet:

Fettdruck

wird für die Namen von Rechnern, für die Benutzernamen und -IDs sowie gelegentlich zur Hervorhebung von Text benutzt.

Kursiv

benutzen wir für Datei- und Verzeichnisnamen, für Befehlszeilenoptionen, E-Mail-Adressen und Pfadnamen sowie zur Hervorhebung von neuen Begriffen.

Nichtproportionalschrift

wird in Beispielen benutzt, um den Inhalt von Dateien oder die Ausgabe von Befehlen darzustellen; außerdem für Umgebungsvariablen und Schlüsselwörter, die in den Programmcode eingebettet sind, sowie für Emacs-Befehle.

Nichtproportionalschrift kursiv

benutzen wir für Optionen, Schlüsselwörter und solche Textstellen, die der Benutzer durch seine eigenen Texte ersetzen muss.

Nichtproportionalschrift fett

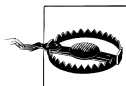
wird in Beispielen für Befehle und solche Textstellen gebraucht, die der Benutzer in genau dieser Form eingeben muss.

KAPITÄLCHEN

weisen auf Schaltflächen und Menüeinträge hin.



Weist auf einen Tipp, einen Vorschlag oder eine allgemeine Anmerkung hin.



Weist auf eine Warnung oder eine Vorsichtsmaßnahme hin.

Kurze Geschichte von Unix

In der Geschichte der Softwareentwicklung kommt es mit *Unix* zu einem Methodenwechsel. Zum ersten Mal wird ein verbreitet eingesetztes Allzweck-Betriebssystem nicht in der spezifischen Maschinensprache des Zielsystems entwickelt, sondern in einer maschinenunabhängigen Hochsprache, was die Portierung (Anpassung) auf andere Maschinen begünstigt. Mit *Unix* bildet zum ersten Mal die Zusammenfassung von

- Betriebssystem,
- dem Entwicklungssystem für das Betriebssystem und den Anwendungen,
- dem Quelltext für alle Komponenten
- und den Anwendungen

eine Einheit. Daraus folgt, dass es sehr einfach ist, Komponenten auszutauschen und bei Vorhandensein des Original-Quellcodes anzupassen.

Von Multics zu Unix

General Electric (GE) und das Project MAC des *Massachusetts Institute of Technologie* (MIT) begannen 1962, aufbauend auf Vorarbeiten am MIT aus dem Jahr 1961, ein Time-Sharing-System zu konstruieren. Time-Sharing-Systeme waren damals neu und boten einen bis dahin noch nicht bekannten Komfort. Programmierer eines Time-Sharing-Systems brauchten nicht mehr ihre Programme und Daten auf Lochkarten zu stan-

zen und dann dem als Dispatcher für das Batch-System arbeitenden Menschen auszuhändigen, nur um dann Tage später das Resultat ausgehändigt zu bekommen und feststellen zu müssen, dass ein winziger, äußerst ärgerlicher Programmierfehler die Arbeit von Wochen zunichte gemacht hatte und eine große Menge Papier verschwendete und hohe Kosten verursachte.

Das Projekt MAC, später wurde daraus das *Laboratory for Computer Science* (LCS), fand finanzielle Unterstützung von der *Advanced Research Projects Agency*. Im Jahr 1965 kauften die *Bell Labs* der *American Telephone and Telegraph* (AT&T) eine *General Electric GE 645* und beteiligten sich an der Entwicklung von *Multics*. Die Anforderungen an eine Mehrprozessormaschine in einem Time-Sharing-System waren noch nicht bekannt. Daher musste laufend die Hardware geändert werden. Der so eingeschlagene Weg führte zu einer sehr teuren und den Leistungsanforderung der *Bell Labs* nicht gerecht werdenden Maschine. Die *Bell Labs* zogen sich daraufhin im Jahr 1969 aus dem Projekt zurück.²

Mitarbeiter wie *K. Thompson*, *D. Ritchie*, *M. D. McIlroy* und *J. F. Ossanna* wollten aber nicht auf die lieb gewonnenen Vorteile eines Time-Sharing-Systems verzichten und versuchten, mit wesentlich geringerem technischen und finanziellen Aufwand eine ähnlich angenehme Arbeitsumgebung zu schaffen. Ohne die Genehmigung finanzieller Mittel für die Beschaffung von Hardware machten sich *K. Thompson*, *R. H. Canaday*, and *D. Ritchie* daran, einen Entwurf für ein Wunsch-Dateisystem auf dem Papier zu erstellen.

Thompson entwickelte nebenher auch nach das Spiel *Space Travel*, das eine Raketenreise – mit Landungen auf Planeten – durch das Sonnensystem simulierte. Diese Programm entwickelte er zuerst unter *Multics*, portierte es dann nach *Fortran* unter dem Betriebssystem *General Electric Comprehensive Operating System* (GECOS) auf einer *GE-635* und portierte es danach auf den Makroassembler einer damals schon obsoleten, aber wegen eines langfristig laufenden Mietvertrages zur Verfügung stehenden *PDP-7* der *Digital Equipment Corporation* (DEC). Auf diese Weise machte sich *Thompson* mit den Verfahren zur Portierung auf die *PDP-7* vertraut.

Dann begann er, das Wunsch-Dateisystem auf der *PDP-7* zu implementieren. Ein Dateisystem ohne Möglichkeiten damit umzugehen bringt nur geringen Nutzen – also implementierte er eine Art Rumpf-*Unix*, das er immer unter *GECOS* entwickelte und auf die *PDP 7* übertrug. Dieses Vorgehen war sehr mühsam und erforderte jedes Mal die Produktion eines Lochstreifens, der von der *GECOS*-Maschine auf die *PDP-7* übertragen werden musste. Die *PDP-7* besaß weder ein eigenes Betriebssystem noch Plattenspeicher, sondern nur eine Laderoutine für Lochstreifen. Es musste also immer eine vollständige Laufzeitumgebung für das gewünschte Programm erzeugt und mit übertragen werden.

Nachdem *Osanna* mit dem geschickt formulierten Projektantrag, man wolle ja kein Universal-System, sondern eine reine Textverarbeitung implementieren, die Beschaffung einer damals höchst modernen *PDP-11* im Jahr 1970 bewerkstelligen konnte, wurde mit der Portierung begonnen. Diese Portierung erfolgte bis Jahresende – die Maschine war so

² <http://cm.bell-labs.com/cm/cs/who/dmr/hist.html>

neu, dass bis dahin noch keine Festplatte lieferbar war – immer noch auf Papier. Als dann endlich alles beisammen war, hatte sich die Implementierung auf der *PDP-7* so bewährt, dass dann doch der Entwicklung eines Allzweck-Betriebssystems zugestimmt wurde. 1971 wurde als erstes Projekt das roff-Text-Formatierungswerkzeug von der Multics-Sprache *BCPL* in den *PDP-11*-Assembler übersetzt. Dieses Werkzeug wurde dann von der Patentabteilung von *AT&T* zum Erzeugen von Patentanträgen verwendet. Das sofort merkbare Kosten-Nutzen-Verhältnis überzeugte das Management.

Thompson meinte, man müsse auch noch *Fortran* auf der *PDP-7* für *AT&T* bereitstellen. Nach einer Woche gab er das ursprüngliche Projekt auf und entwickelte die Sprache *B*, eine Interpretersprache, die zur Laufzeit in Maschinensprache übersetzt wurde und von *BCPL* beeinflusst war. Wegen der geringen Rechenleistung wurde nach wie vor die Entwicklung auf der *PDP-11* in Maschinensprache und nicht in der Hochsprache *B* vorgenommen. Dies bot 1971 Anlass, die Compilersprache *C* zu entwickeln. Der *C*-Compiler übersetzt in der Hochsprache formulierte Programme in direkt ausführbare Maschinensprache. 1973 wurde dann auch der *Unix*-Kernel in *C* vollständig – bis auf etwas 1000 Zeilen Assemblercode – neu geschrieben. Diese 1000 Zeilen waren ein Tribut an die Leistungsfähigkeit damaliger Computer, bei denen die Optimierung laufzeitkritischer Teile des Betriebssystems unumgänglich war.

Dateibasierte Betriebssysteme

Eine wesentliche Stärke, die *Linux* von seinen Ahnen geerbt hat, ist das Dateisystem. Eine nahe liegende Methode zur Bezeichnung von Geräten, Orten zur Speicherung von Daten, Programmen und so fort ist es, für jede Kategorie eine eigene Klasse von Namen zu schaffen. Bei Erweiterungen eines Computersystems um Programme und Wechsel von einer Computer-Architektur zu einer anderen, muss dann der Inhalt der jeweiligen Kategorie geändert werden.

Wenn dagegen jedes in Betracht kommende Objekt, wie Gerät, Datei und Programm direkt eine Repräsentanz im Dateisystem hat, so ist eine Änderung oder Anpassung auf eine geänderte oder gar eine vollkommen andere Plattform sehr einfach vorzunehmen. Geräte werden durch symbolische Namen repräsentiert. Dadurch ist eine Änderung in der Hardware transparent für den Benutzer.

Der Austausch verschiedener Versionen von Programmen kann so bewerkstelligt werden, dass das neue Programm an die Stelle im Dateisystem platziert wird, die das alte innehatte. Eine andere Möglichkeit ist es, das neue Programm an einer Stelle zu hinterlegen, die früher als die Stelle des alten Programms durchsucht wird, da zur Ausführung eines Kommandos das Dateisystem an vorher festgelegten Stellen in genau festgelegter Reihenfolge durchsucht wird. Die Suchreihenfolge kann frei eingestellt werden.

Die Grundideen von Unix

Unix wurde für kleine Computer entwickelt, die mehrere Benutzer scheinbar gleichzeitig, in Wirklichkeit schnell hintereinander bedienen können. Linux trennt die verschiedenen Benutzer in eigenständigen Sitzungen, die sich gegenseitig nicht beeinträchtigen können. Den verschiedenen Benutzern werden abgegrenzte Arbeitsbereiche zugewiesen, sie können aber auch gemeinsame Arbeitsbereiche schaffen. Alle Ressourcen wie Programme, Plattenplatz, Druckerzugriff und Rechenzeit stehen grundsätzlich jedem Benutzer zur Verfügung. Die Systemverwalter können jedem Benutzer für die Verwendung bestimmter Ressourcen gezielt Einschränkungen auferlegen oder ihm die Nutzung vollständig untersagen.

Es gibt zwei Kategorien von Benutzern auf einem *Unix*-System – die gewöhnlichen Benutzer und die Systemverwalter (*root*). Nur die Systemverwalter dürfen direkt auf Geräte zugreifen, Konfigurationen ändern und gewöhnliche Benutzer verwalten. Die gewöhnlichen Benutzer dürfen die restlichen Leistungsmerkmale eines *Unix*-System nutzen.

Unix ist strikt dateibasiert. Zum Beispiel wird ein Druckvorgang dadurch vorgenommen, dass die Druckdaten in die Datei hineinkopiert werden, die die Druckerschnittstelle repräsentiert.

Unix besteht aus vielen kleinen Programmen und Werkzeugen, die beliebig kombiniert werden können. Am Beispiel der Verarbeitung von elektronischer Post (E-Mail) zu Ausdrucken kann das Ineinandergreifen gut beschrieben werden. Ein Programm setzt einen Datenfluss in Bewegung (E-Mail beim Provider abholen, *fetchmail*), ein weiteres filtert ihn (Zuordnung zu den Benutzern, *sendmail*), dann werden die Daten individuell gespeichert oder weitergeleitet (*procmail*). Die gespeicherten Daten werden wieder in Bewegung gesetzt (*mailq*), dann formatiert (*a2ps*) und ausgegeben (*lp*). Wenn der Drucker nicht *postscriptfähig* ist, erfolgt im Druckprozess eine weitere Filterung und Formatierung (*ghostscript*)

Das Verhältnis zwischen Unix und seinen Nachfahren

Aus kartellrechtlichen Erwägungen – wegen eines möglich erscheinenden Anti-Trust-Verfahrens – gab *AT&T* ihr Produkt *Unix* nicht an kommerzielle Endkunden, sondern nur an Universitäten – einschließlich des Quellcodes – weiter. Das amerikanische Verteidigungsministerium erkannte die Qualitäten des *Unix*-Systems und beauftragte über seine Wissenschafts-Förderungs-Agentur *ARPA* (*Advanced Research Projects Agency*, jetzt *DARPA*, *Defense Advanced Research Projects Agency*) die *University of California*, ein vollständiges Betriebssystem zu erstellen, das zum *Original-Unix* kompatibel ist, um nicht nur von einem Hersteller abhängig zu sein. Die *University of California at Berkeley* (*UCB*) nutzte diese Möglichkeit – von der *ARPA* finanziell unterstützt – für fast jeden Bestandteil von *Unix* eine leistungsgesteigerte Eigenentwicklung bereitzustellen. Zur der Zeit, als *AT&T* die *Unix*-Abteilung aus dem Mutterkonzern ausgliederte, kam es in den

Jahren 1992 bis 1994 zu einem Rechtsstreit zwischen den *Unix System Laboratories* (USL) und der gesamten *University of California*, der in einen Vergleich mündete. Danach gingen *Unix* und das Derivat der damals gegründeten *Berkeley Software Development Inc.* (BSDI) vollständig eigene Wege. Heute trifft man gelegentlich das kommerzielle *Unix*, das frei verfügbare *BSD*-Derivat, aber vor allem *Linux* an. Zuletzt fand dann doch eine Zerschlagung von *AT&T* wegen der Beherrschung des amerikanischen Telephonmarkts durch *AT&T* statt.

Vor dem Anti-Trust-Verfahren gegen *AT&T* wurde *USL* von *Novell* gekauft. So entwickelte sich das kommerzielle *Unix* – von *AT&T* zu *Unix Software Laboratories* (USL), dann zu *Novell* und zuletzt zu *SCO*.

Die Entwicklungen der *Universität of California at Berkeley* wurden unter einer besonderen Urheberrechtsklausel veröffentlicht, die eine beliebige Verwendung des Quellcodes erlaubt. Zusammen mit den Erzeugnissen der *FSF*³ (*Free Software Foundation*, Gründer *Richard Stallmann*), die eine Weiterverbreitung stärker reglementieren, konnten weitere *Unix*-artige Systeme wie *Minix* von *Andrew Tanenbaum* erzeugt werden. Die Programme der *FSF* werden *GNU*⁴ (*GNU is not Unix*) genannt. *Linus Torvalds* entwickelte dann *Linux* auf der Basis von *Minix*. Die Produkte der *FSF* dürfen nur in unveränderter Form oder zusammen mit dem Quelltext der selbst vorgenommenen Änderungen weiterverbreitet werden.

Die Geschichte von Linux

Linus Torvalds begann im Jahr 1991 auf Basis von *Minix* mit einer Eigenentwicklung zu experimentieren. Das damalige *Minix* war auf die erste Generation von Personal-Computern ausgerichtet – es unterstützte keine hardwarebasierten Mechanismen zur Speicherverwaltung, die die moderneren Prozessoren bieten – sondern verwirklichte eine Software-Speicherverwaltung. Ein solches Vorgehen erlaubt es zwar, einen modernen Ansatz in der Speicherverwaltung auf einem in die Jahre gekommenen Computer vorzuführen, hat aber den Nachteil, sehr viel Rechenleistung und damit Rechenzeit für die interne Verwaltung zu verbrauchen. Eine Hardware-basierte Speicherverwaltung verlangsamt die Ausführung eines Programm wesentlich weniger.

Nach Beherrschung der Speicherverwaltung, einer der komplexesten Aufgaben im Bereich Systemdesign, war es nahe liegend, gleich ein komplettes System zu erstellen. *Linus Torvalds* wollte dabei die Vorteile von etablierten technischen Standards nutzen und hielt sich an die *POSIX*-Spezifikationen⁵ und Veröffentlichungen wie *Mauritz J. Bachs: The design of the Unix operating system*, wohl wissend, dass damit die Verfügbarkeit von Standard-Werkzeugen, wie dem *GNU-C-Compiler* und danach den *GNU-Werkzeugen* sichergestellt war.

³ <http://www.fsf.org/>

⁴ <http://gnu.org/>

⁵ <http://www.unix.org/version3/>

Zur gleichen Zeit wurde in Berkeley vom Ehepaar *Jolitz* das 386BSD, ein Abkömmling der BSD-Reihe speziell für die PC-Architektur vorgestellt. Dieser Ansatz geriet jedoch ab 1992 in die drei Jahre mahelnden Mühlen des Rechtsstreits zwischen dem *Unix Software Laboratory (USL)*, dem damaligen Rechteinhaber von *Unix* und der *University of California*. *Linux* war von diesen Auseinandersetzungen nicht betroffen und hatte zudem für die Teilnehmer den Vorteil, dass *Linus Torvalds* sich eines anderen Ansatzes als die Kalifornier bediente – man musste keine formale Anerkennung erwerben, sondern jeder konnte seine Ideen an den Verwalter des Code-Baumes, *Linus Torvalds* schicken. Wenn der Patch vernünftig schien, wurde er eingepflegt, wenn nicht, gab es eine Begründung warum. Dieser Ansatz gilt mehr oder weniger heute noch, wenngleich *Linus Torvalds* nicht mehr alle Entwicklungslinien des Kernels pflegt und fast alle Teilaufgaben der Software-Revision und –Pfleger delegiert hat. *Eric Raymond*⁶ hat in seinem Aufsatz *The cathedral and the basar*, sinngemäß übersetzt mit *hierarchische gegenüber selbstorganisierter Aufgabenverteilung* die Vorgehensweise beschrieben.

Die konsequente Anwendung der striktesten Version *GNU Public License*⁷ (*GPL*) auf den *Linux*-Kernel und alle abgeleiteten Betriebssystem-Bestandteile machte es unmöglich, sich durch Patentansprüche, Warenzeichenrechte und Urheberrechtsansprüche bestimmte Teile des *Linux*-Kernels und den Bestandteilen des Betriebssystems anzueignen und damit mögliche Mitbewerber und Anwender von der Nutzung auszuschließen oder die Nutzung kostenpflichtig zu machen. In der Folge entstanden viele neue Firmen, die nicht wie bisher die Nutzungsrechte an einem Betriebssystem verkauften, sondern die Dienstleistung anboten, ein frei verfügbares Betriebssystem gebrauchsfertig zu konfektionieren, auf Datenträger zu bringen und Support anzubieten. Dies spiegelt sich auch in der Bezeichnung Distributoren für diese Firmen wider.

Die Nutzer von Linux

Die Entwicklung von *Linux* und seinen Anwendungen ist heute so weit gediehen, dass jeder Benutzer seine Anforderungen durch Produkte oder eine Systemleistungen von *Linux*-Computern abgedeckt finden kann. Ein jeglicher, der einen Computer für seine Arbeit oder sein Vergnügen einsetzt, kann auch einen *Linux*-Computer nutzen. Die Zahl der *Linux*-Nutzer nimmt stetig zu und hat mittlerweile eine Größe erreicht, die es für die klassischen Anbieter von Dienstleistungen interessant macht. So kann der Nutzer oder potenzielle Nutzer Dienstleistungen einkaufen und frei aus einem Warenkorb auswählen, was er denn in Anspruch nehmen möchte.

⁶ <http://www.catb.org/~esr/>

⁷ <http://www.gnu.org/copyleft/gpl.html>

Die Einsatzgebiete von Linux

Die übliche Zweiteilung der am weitesten verbreiteten EDV-Struktur in Client und Server wird durch den Einsatz von Linux verwischt. *Linux* kann einerseits gleichzeitig als Client und Server in Betrieb sein, ohne die sonst gewohnten Nachteile, wie Leistungseinschränkung, lizenzrechtliche Regelungen und gegenseitige Beeinflussung von Programmen und Dienstleistungen in Kauf nehmen zu müssen. Die Client- und Servertätigkeit kann aber auch strikt getrennt werden, sowohl auf einem *Linux*-Computer als auch zwischen verschiedenen *Linux*-Computern. *Linux* kann vom plattenlosen Grafik-Terminal für Terminal-Server-Anwendungen bis zum schwergewichtigen Datenbankserver mit allen nur denkbaren Sicherheitsmechanismen maßgeschneidert werden.

Mit *Linux* kann

- eine klassische Büroarbeitsumgebung geschaffen,
- ein Datenbankserver realisiert,
- ein Web-Server im Internet verwirklicht,
- ein Datei- und Druckserver für jede denkbare Netzwerkkumgebung gefunden,
- Firewall-, Proxy- und Sicherheitslösungen geschaffen,
- und natürlich auch gelernt und gespielt

werden.

Die Gründe für den Einsatz von Linux

Eine Entscheidung für *Linux* ist eine Entscheidung gegen eine Monokultur auf der Anbieterseite. Es gibt viele Anbieter von *Linux*-Distributionen mit ganz unterschiedlichen Profilen, Kosten und Angeboten. Technisch qualifizierte, belesene oder lese- und lernfreudige *Linux*-Anwender können auch auf jegliche fremde Dienstleistungen völlig verzichten und das gesamte System selber erstellen und warten. Alles ist umfassend von dem *Linux Documentation Project*⁸ dokumentiert

Die Produktgestaltung unterliegt bei *Linux* nicht den Entscheidungen eines einzigen *Linux*-Herstellers. Der Anwender kann nach seinen Anforderungen und Budget-Vorstellungen entscheiden, welche Produkte und Dienstleistungen er aus einem Baukasten auswählen möchte. Bei diesem Baukasten gibt es für jede Funktion mehr als ein Angebot. Das im Segment der marktbeherrschenden Softwareanbieter übliche – bei manchen Anbietern sogar nachträgliche – Einführen kostentreibender, technischer Leistungseinschränkungen zur Durchsetzung weiterer finanzieller Forderungen ist bei einem *Linux*-System nicht durchführbar. Stattdessen ist wegen der Vielfältigkeit des Angebots der Kunde König.

⁸ <http://www.tldp.org/>

Der Quelltext (Source Code) für den Kernel und das Betriebssystem ist frei verfügbar. Mehrere, voneinander unabhängige Organisationen, zum Teil kommerzieller, zum Teil wissenschaftlicher, zum Teil amtlicher Natur nehmen regelmäßig Qualitätskontrollen (Code-Reviews) des Quelltexts vor, so dass Fehler schnell entdeckt werden. Es ist eine große Anzahl von freien und angestellten Programmierern mit Linux und verbundenen Projekten beschäftigt, die natürlich auch entsprechende Aufmerksamkeit auf den Quelltext richten.

Die freie Verfügbarkeit des Quelltextes von Linux und seinen Werkzeugen unter der *GNU Public License*⁹ (*GPL*) ermöglicht es einem Linux-Anwender, eine Aufgabe, die durch Weiterentwicklung der Standardbestandteile erledigt werden kann, einem Dienstleistungsanbieter zur Lösung anzutragen. Die Entwicklungsarbeit wird dadurch verringert. Wenn zur Programmierung *GPL*-Werkzeuge und der *GPL* unterliegende Bestandteile von *Linux* verwendet werden, muss der Quellcode ausgeliefert werden. Sollte dann dieser Anbieter für ein Teilproblem keine Lösung anbieten können, kann der Anwender dieses Teilproblem an einen dritten Anbieter übergeben. Wenn irgendwann der ursprüngliche Anbieter nicht mehr zur Verfügung stehen sollte, kann der gesamte Code einem neuen übergeben werden.

Die Gründe für den Einsatz von Linux in Firmen

Im Firmeneinsatz kann sich *Linux* als universell einsetzbares Betriebsmittel qualifizieren. Der Aufwand für Mitarbeiterschulung kann so optimiert werden. Das gleiche Grundsystem kann für Arbeitsplatz-Computer, für Abteilungsserver und für den Gesamtbetrieb als Enterprise-Resource-Planning-System (ERP) eingesetzt werden. Es gibt unter anderem *SAP* für *Linux*. Für *Linux* sind auch die gängigen Werkzeuge im Bereich Konstruktion, Messwesen und Produktionssteuerung verfügbar.

Linux-Computer können so konfiguriert werden, dass keinerlei Missbrauch möglich ist. Beschaffungs- und Folgekosten sind leicht und sicher zu begrenzen. Eine Anbindung an vorhandene »mittlere Datentechnik« ist möglich und eine Migration von Anwendungen und Daten wird durch bereitgestellte Werkzeuge erleichtert. *Linux* kann auch für Infrastrukturmaßnahmen wie Firewall, Router, Proxy und Bridging verwendet werden. Für autonome Systeme und Maschinensteuerungen (Embedded Systems) kann *Linux* als Entwicklungs- und Zielplattform Verwendung finden.

Die Gründe für den Einsatz von Linux für öffentliche Einrichtungen

Im öffentlichen Bereich ist die Datensicherheit immer ein besonders kritischer Faktor. Anwendungen auf Basis von *Linux* können besonders gut absichert werden, da alle Bestandteile von *Linux* gut dokumentiert und anpassbar sind. Nicht benötigte Bestandteile von Linux und seinen Anwendungen können entfernt werden.

⁹ <http://www.gnu.org/gpl/>

Im öffentlichen Bereich finden häufig Veränderungen in Vorschriften, Richtlinien und Gesetzen statt, die Anpassungen von Daten und Programmen erforderlich machen. *Linux* erlaubt den Einsatz von zentraler Systemadministration, um die systematische Verteilung obligatorischer Aktualisierungen vorzunehmen.

Notwendige Wartungsmaßnahmen können beim Einsatz von *Linux* auf einen Zeitpunkt gelegt werden, der den Anforderungen an die Haushaltsregeln genügt.

Die Gründe für den Einsatz von Linux für Privatanutzer

Ein *Linux*-basierter Arbeitsplatzcomputer für zu Hause kann sehr kostengünstig beschafft werden.

Die Standard-*Linux*-Distributionen enthalten alle Werkzeuge, die im kleinen Büro Anwendung finden – Textverarbeitung, Tabellenkalkulation und Datenverwaltung.

Mit *Linux* steht ein System zur Verfügung, das für jede gebräuchliche Programmiersprache mindestens einen Compiler ohne Zusatzkosten zur Verfügung stellt. *Linux* ist mit seinen Werkzeugen ideal als Lern- und Lehrplattform. Das System selbst kann risikolos verändert werden, um die Auswirkungen der Änderungen zu testen.

Systemeigenschaften und -ausgestaltung

Der Entwurf von *Linux* lehnt sich stark an den Entwurf von *Unix System V* an, und verwirklicht zudem die meisten Eigenschaften von *BSD 4.4*. Programme und Entwicklungssysteme können meist durch einfache Neuübersetzung von den Ursprungsplattformen *System V* und *BSD* auf *Linux* übertragen werden. Für Programme, die unter der *Intel*-Version von *System V* laufen, gibt es außerdem einen Emulator, der auf der Intel Binary Compatibility Specification (*IBCS*¹⁰) basiert.

Der Entwurf von *Linux* folgt konsequent den *POSIX*-Spezifikationen¹¹. Damit sind alle Programme, die auf diesen Spezifikationen aufbauen, für *Linux* verfügbar zu machen. Die *GNU*¹²-Software basiert auf den *POSIX*-Schnittstellen. Die Konformität mit den Standards ist zertifiziert.

Was mit dem Einsatz von Linux erreicht werden soll

Der wesentliche Unterschied von *Linux* zu bekannten kommerziellen Systemen ist seine Anpassbarkeit, Vielseitigkeit, Transparenz und Verfügbarkeit.

¹⁰ Intel Binary Compatibility Standard, Intel, MacGraw-Hill 1992

¹¹ <http://www.posix.org/>

¹² <http://www.gnu.org/>

Linux-basierte Computer können fast vollständig auf die Bedürfnisse einer bestimmten Anwendung optimiert werden. Die Massenspeicherverwaltung kann jeder bekannten Bauform entsprechen. Mehr-Prozessor-Hochleistungssysteme sind genauso möglich wie rein auf Festkörperspeicher (Read Only Memory, ROM) basierende, stromsparende Kontrollsysteme.

Linux-Computer können für Kassensysteme (POS, Point of Sales) in Handelsunternehmen verwendet werden wie für wissenschaftliche Verbundsysteme (Grid Computing). Grafische Arbeitsplätze können ebenso realisiert werden wie Abrechnungssysteme für Copy-Shops, Gebäudeleittechnik auf höchstem Niveau wie die Steuerung und Überwachung von Industrieanlagen.

Die Verfügbarkeit und Offenlegung aller Systemkomponenten erlaubt es den Anwendern, das Einschmuggeln von Beobachtungsfunktionen (Spyware) auszuschließen. Eine gezielte Begrenzung der Leistung von Programmen eines *Linux*-Systems auf Benutzer- und Prozessoranzahl und die Beschränkung der Ausführbarkeit auf einen bestimmten Zeitraum ist technisch möglich, aber nicht üblich. Beschränkungen solcher Art sind nur für bestimmte, einige wenige kommerzielle Software-Komponenten üblich und nicht für das Gesamtsystem möglich.

Die Verfügbarkeit des Quelltextes macht es möglich, unternehmenskritische Anwendungen auf die jeweils verfügbare oder notwendige Hardware anzupassen, unabhängig von der Verfügbarkeit von bestimmten, möglicherweise obsolet gewordenen Baureihen und Bausteinen. Außer durch Patentklagen kann den Anwender kein Interessenvertreter mehr daran hindern, einmal getätigte Investitionen weiter zu nutzen.

Die benötigte Hardware

Die benötigte Hardware hängt von der verwendeten *Linux*-Distribution ab. Sie reicht von Distributionen, die speziell für Industrierechner auf Basis eines stromsparenden *Intel*-386 ausgelegt sind bis zu Mainframeversionen mit Dutzenden von Höchstleistungsprozessoren. Der durchschnittliche Endanwender wird einen Standard-PC verwenden, der auf einer *Intel*-Architektur¹³ aufbaut, der ab 128 MB Hauptspeicher (Random Access Memory, RAM) zur Verfügung hat und eine Festplatte mit freiem Platz ab 40 GB besitzt. Der Anwender sollte möglichst viel Hauptspeicher vorsehen, dafür belohnt ihn dann *Linux* auch mit hoher Rechenleistung. Eine handelsübliche Grafikkarte, eine Tastatur, eine Maus und ein CD- oder DVD-ROM-Laufwerk vervollständigen die notwendige Ausrüstung. Die Details und eventuell vorhandene oder wünschenswerte Hardware oder Geräte beschreiben die Distributoren detailliert in ihren Hardware-Datenbanken¹⁴.

¹³ Hersteller sind <http://www.intel.com/>, AMD <http://www.amd.com/>, VIA <http://www.via.com.tw/>, IBM <http://www.ibm.com/> und andere.

¹⁴ <http://hdb.suse.de/>

Eigenschaften der Software

Jede *Linux*-Distribution bietet Werkzeuge, meist von der *Free Software-Foundation* (FSF), zur Dateiverwaltung (*mv*, *ren*, *rm*, *mkdir*...) und Bearbeitung von Text-Dateien (*ed*, *vi*, *emacs*, *joe*...). Das älteste und ursprünglichste Werkzeug zur Textverarbeitung, der Formatierer *roff* ist Bestandteil des *Linux*-Hilfesystems. Weiter fortgeschritten ist das Satzsystem *T_EX* und sein SGML-Makropaket *L^AT_EX*. Es gibt zudem das *GNU*-Info-System zur Beschreibung der *GNU*-Werkzeuge. Die *GNU*-Werkzeuge bieten Shell (zeilenorientierter Kommando-Interpreter), Werkzeuge zur Zeichenkettenmanipulation (*find*, *grep*, *join*, *cut*...). Viele Werkzeuge sind auch im für Browser geeigneten HTML-Code dokumentiert.

Die nächste Stufe bilden Tabellen- und Rechenprogramme wie *gnumeric* und *bc*, mit deren Hilfe komplexe Berechnungen erledigt werden können. Interpretersprachen wie *awk* und *perl* bieten sehr leistungsfähige Werkzeuge, um beispielsweise Abrechnungsdaten für Telefon, Strom und Heizung auszuwerten. Diverse kleinere Versorger bedienen sich tatsächlich dieser Werkzeuge.

Auf der Basis grafischer Benutzeroberflächen gibt es klassische Büro(»Office«)-Arbeitsumgebungen, die über komfortable Import- und Exportfunktionen mit anderen Arbeitsumgebungen und Systemen zusammenarbeiten können.

Von *Unix* hat *Linux* den klassischen Ansatz geerbt, in sich geschlossene, verknüpfbare und standardisierte Anwendungen bereitzustellen. Mit solchen Anwendungsbausteinen können bisher nicht gekannte Aufgaben, auch solche von hoher Komplexität, durch einfaches Zusammensetzen, Hintereinanderschalten und das Einschalten von Skriptsprachen erledigt werden. Wenn eine passende Spezial-Lösung wünschenswert ist, kann bis dahin die Aufgabe unter Inkaufnahme von einem gewissen Rechenzeitbedarf erledigt werden.

Bezugsquellen von Linux

Durch das gewünschte Medium wird die Auswahl der Bezugsquellen für *Linux* bestimmt. *Linux* kann auf Disketten, CD-ROMs, DVD-ROMs und aus dem Internet bezogen werden. Das Internet war das erste Medium für *Linux*. Es wird immer noch intensiv genutzt und stellt für andere Verbreitungsmedien die Möglichkeit zu zeitnaher Verbreitung von Aktualisierungen dar. Disketten wurden als zweites weit verbreitetes Medium eingesetzt und sind kaum noch im Gebrauch. Sie können anstelle von CD-ROMs verwendet werden, wenn keine CD-ROMs gebrannt werden sollen oder können. Vorlagen (Images) für CD- und DVD-ROMs können sowohl aus dem Internet heruntergeladen als auch im Handel erworben werden. Im Handel gibt es sowohl CD- und DVD-ROMs, die von Distributoren stammen als auch CD- und DVD-ROMs, die so genannte Abzüge enthalten, das sind inhaltlich genaue Kopien von Servern im Netz.

Kommerzielle Versionen

Für kommerziell orientierte Distributionen sind zwei Verbreitungsformen üblich – DVD oder CD-ROM und Herunterladen aus dem Internet. Die Server für das Herunterladen stellen die Daten in einem passwortgeschützten Bereich zur Verfügung. Neben den Daten stellen viele kommerzielle Distributoren eine fein abgestufte Palette von zusätzlichen Dienstleistungen zur Verfügung. Manche Enterprise-Server werden auch für Mainframe-Architekturen angeboten.

Red Hat¹⁵

Es werden kommerzielle Versionen für Server-Dienste und technische Workstations (*Red Hat Enterprise Linux*) und eine Desktop-Version (*Red Hat Desktop*) für Kunden mit vielen Arbeitsplätzen geliefert. Sie sind über *Red Hat* direkt wie auch über Partner erhältlich. Für die Produkte stehen unterschiedliche Support-Varianten zur Verfügung. Das automatische Update der Software über das *Red Hat Network* ist immer eingeschlossen. *Red Hat* bietet zur Zeit kein kommerzielles, Endkunden-orientiertes Box-Produkt für Einzel-Arbeitsplätze an. *Red Hat* unterstützt aber die freie *Fedora*-Distribution, die auch von Dritten auf Datenträger gebracht und verkauft wird. Von *Red Hat*¹⁶ werden unterschiedliche Versionen der *Fedora*-Distribution kostenfrei zum Download bereitgestellt.

SuSE¹⁷

Aus Nürnberg sind ein Box-Produkt für Endkunden, *SuSE Linux Professional*, eine Serverlösung, *SuSE Linux Enterprise Server*, und eine Arbeitsplatzlösung für Großkunden, *SuSE Linux Desktop* erhältlich. Alle Produkte werden auf Datenträgern mit gedruckter Dokumentation ausgeliefert. Der Verkaufspreis für das Box-Produkt enthält immer eine befristete Grund-Wartung, zusätzliche Wartung muss als Einzelvertrag vereinbart werden. Die Endkundenlösung, *SuSE Linux Professional*, ist nach einer gewissen Zeit nach Auslieferung des Box-Produktes immer auch im Internet gebührenfrei erhältlich. Für diese Version gibt es keinerlei Unterstützung.

Knoppix¹⁸

Die *Knoppix*-Distribution spielt als kommerzielle Distribution eine Sonderrolle. Die im Internet und sonst wie verbreitete *Knoppix*-Version wird nur als Beta und ohne jeglichen Support vertrieben. Qualifizierte Fehlermeldungen sind aber hoch willkommen. *Hans Knopper* stellt projektspezifische Sonderkonfektionierungen der *Knoppix*-Distribution her, die mit Wartung verkauft werden. Es gibt Händler, die Support für die Standard-Version anbieten. *Knoppix* ist so angelegt, dass es ohne jegliche Installation auf die Fest-

¹⁵ <http://www.redhat.com/>

¹⁶ <http://fedora.redhat.com>

¹⁷ <http://www.suse.de/>

¹⁸ <http://www.knopper.net/>

platte nur von CD- oder DVD-ROM laufen kann. Es kann als vollwertige Linux-Distribution verwertet werden, und wird außerdem als Rettungs-, Reparatur- und Viren-Beseitigungs-System verwendet. Die *Knoppix*-Distribution wird vom *Bundesamt für Sicherheit in der Informationstechnik (BSI)*¹⁹ verwendet.

Nicht-kommerzielle Versionen

Für die nicht-kommerziellen Versionen ist das Internet das primäre Verbreitungsmedium. Es werden sowohl einzeln auswählbare Pakete, das sind gruppierte Zusammenstellungen von Programmen und Dateien angeboten, als auch ISO-Images. Die ISO-Images sind bitgetreue Abbilder von CD-ROMs im ISO-9660-Format, die mithilfe von Brennprogrammen²⁰ auf CD-R-Datenträger gespeichert werden können. Einige Händler bieten auch konventionell erzeugte CD-ROMs an, die gelegentlich sogar mit Dienstleistungen, aber vom Händler, nicht vom Erzeuger der Distribution angeboten werden.

Debian²¹

Die *Debian*-Distribution unterliegt der strengsten Auslegung von Free Software im Sinne der *FSF*. Kommerzielle Programme werden nicht eingebunden, sind aber nachträglich integrierbar. Schwerpunkte sind hohe Sicherheit, Codestabilität und Unterstützung möglichst vieler Hardware-Plattformen.

Gentoo²²

Die *Gentoo*-Distribution ist auf höchstmögliche Aktualität ausgelegt. Es kann nur ein Grundsystem als ISO-Image heruntergeladen werden, alle weitere Software wird danach als Quelltext geladen und übersetzt (compiliert). Die laufende Aktualisierung beruht im Gegensatz zu anderen Distributionen nicht auf der Bereitstellung von direkt ausführbaren Paketen, sondern über die Verbreitung des Quelltextes. Für jeden Computer, auf dem *Gentoo* im Einsatz ist, können spezifische Optimierungen eingetragen werden, dadurch kann wesentlich schneller ausführbarer Code erzeugt werden, als dies für die anderen Distributionen möglich ist. Die *Gentoo*-Distribution verlangt nach einer Flatrate für den Internetzugang.

Herunterladen

Das Herunterladen kann auf jedem Computer erfolgen, der einen Internetzugang besitzt und genug Plattenplatz zur Verfügung hat. Auf einem Computer, der weder unter *Linux* oder einem *Unix*-Derivat läuft, wird sinnvollerweise mindestens ein passendes ISO-

¹⁹ <http://www.bsi.bund.de/>

²⁰ zum Beispiel ISO-Recorder von Alex Feinman <http://isorecorder.alexfeinman.com/isorecorder.htm>

²¹ <http://www.debian.org/>

²² <http://www.gentoo.org/>

Image für eine CD-ROM heruntergeladen und dann eine bootfähige CD-ROM gebrannt. Zur weiteren Installation wird dann nach dem Start von *Linux* entweder das Internet verwendet oder es werden die übrigen Daten von weiteren bereitgestellten CD-ROMs installiert. Aktuelle *Linux*-Distributionen mit Grafikerunterstützung verwenden fünf oder mehr CD-ROMs

ftp.gwdg.de

Die *Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen* stellt einen großen FTP-Server²³ zur Verfügung, auf dem fast alle *Linux*-Distributionen im Verzeichnis */pub/linux* erhältlich sind. Kommerzielle Distributoren lassen dort auch ihre Aktualisierungen (Updates) ablegen. Das Herunterladen kann sowohl mit einem Browser als auch mit speziellen FTP-Programmen vorgenommen werden. Ideal sind FTP-Programme wie *wget*²⁴, die ein automatisches Wiederaufnehmen von begonnenen, aber unterbrochenen Herunterladevorgängen erlauben.

www.kernel.org

Die zuverlässigste Quelle für Kernel und verbundene Werkzeuge ist der Server²⁵ der Organisation *kernel.org Inc.* aus Kalifornien. Zuerst hier werden die neuesten Kernel-Versionen zur Verfügung gestellt. Insbesondere bei *Linux*-Computern, die kritische Daten halten oder eine wichtige Aufgabe erfüllen, muss regelmäßig nachgesehen werden, ob eine neue Kernelversion mit Änderungen bereitgestellt wurde, die die Aufgaben des *Linux*-Computers betreffen.

Auswahlkriterien für die zu wählende Installation

Es gibt mittlerweile um die 400 *Linux*-Distributionen²⁶. Das Spektrum reicht von Allzweck-Distributionen²⁷ über Distributionen für bestimmte Sprachen²⁸, die nicht das lateinische Alphabet benutzen bis zu Distributionen für Echtzeitanwendungen²⁹. Der Durchschnitts-Anwender wird jedoch mit einer Durchschnitts-Distribution sehr zufrieden sein, da die Spezialisierung der Distribution auch immer einen gewissen Komfortverlust für den Anwender bewirkt.

²³ <ftp://ftp.gwdg.de/>

²⁴ <http://www.gnu.org/software/>

²⁵ <http://www.kernel.org/> und <ftp://ftp.kernel.org/>

²⁶ <http://www.linux.org/dist/index.html>

²⁷ <http://www.redhat.com> <http://www.suse.de/>

²⁸ <http://ivrix.org.il> hebräisch <http://www.haydarlinux.com/> arabisch <http://cle.linux.org.tw> taiwanesisch <http://www.redflag-linux.com/eindex.html> festland-chinesisch

²⁹ <http://www.linuxworks.com/> <http://www.rtlinux.org/>

Anfänger

Für den Anfänger ist eine Distribution zu empfehlen, die neben vollständigen Datenträgern auch gedruckte Information bereitstellt. Eine automatisierte Installation, wie sie etwas *SuSE* oder *Red Hat Linux* bereitstellen, ist von Vorteil. Dem Anfänger wird durch einen auf die Distribution optimal abgestimmten Computer sehr geholfen. Insbesondere Grafik-Karten unterliegen einem äußerst schnellen technischen Wandel. Die Verfügbarkeit von bezahlbarem oder im Kaufpreis inbegriffenen Installationssupport hilft, selten auftretende, dann aber sehr frustrierende Fehlversuche zu vermeiden.

Fortgeschrittene Nutzer

Wer seine Wünsche kennt, kann sich sehr zielgerichtet die Distribution aussuchen, die seiner Wunschvorstellung am nächsten kommt. Häufige Wünsche sind möglichst aktuelle Programmpakete fürs Büro, Grafikprogramme wie *gimp* oder eine besonders neue *KDE-Oberfläche (K-Desktop-Environment)*. Die Benutzerfreundlichkeit ist nur eines von mehreren Auswahlkriterien, und der Benutzer wechselt häufiger sowohl innerhalb der Distributionen einer Kategorie als auch zwischen den kommerziellen und den freien Distributionen. Fast alle Distributionen erlauben es, eine fremde Distribution gewissermaßen zu aktualisieren. Wenn grundlegende Konfigurationsmechanismen gemeinsam sind, werden sie auch übernommen. Sonst müssen die Grundeinstellungen angepasst oder wiederholt werden.

System-Administratoren (sicherheitsbetonte Distributionen)

Für Systemadministratoren ist die Aktualität der Software nur insofern interessant, als die Software keine bekannten Fehler mehr enthalten darf. Dagegen ist neben einem Kernel, der allen notwendigen Sicherheitsmaßnahmen umsetzt, oft die Verträglichkeit mit großen kommerziellen Softwarepaketen maßgebend. Datenbankhersteller wie *Oracle* und *IBM* und Softwarehäuser wie *SAP* zertifizieren bestimmte Distributionen, die dann genau nach Herstellerangaben mit Patches (Aktualisierungen und Anpassungen) zu versehen sind. Auf *Linux*-Computern, die für eine Firma oder eine Einrichtung wichtige Dienste versehen, wird man sinnvollerweise eine Distribution einsetzen, deren Betreuer garantierte Antwortzeiten auf Fehlermeldungen gewährleistet.

Namensvergabe

Ein *Linux*-Computer, der weder das Internet nutzt noch Teilnehmer in einem lokalen Netzwerk ist, braucht sich wegen der Vergabe von Namen an keine besonderen Regeln zu halten. Wenn jedoch die Anbindung an ein Computernetzwerk erfolgen soll, müssen dessen Richtlinien und Regeln eingehalten werden, da sonst der *Linux*-Computer nicht angesprochen werden kann, er selbst keinen ansprechen kann oder der Betrieb des Netzwerks empfindlich gestört wird (beispielsweise bei mehrfacher Vergabe des gleichen Namens).

Regeln für die Namensvergabe

Der typische Name eines *Linux*-Computers

mein.linux.comp

hat drei Bestandteile:

1. der Computername *mein*
2. der Name des lokalen Netzes *linux* und
3. die Top-Level-Domain *comp*

Die Top-Level-Domain und der Netzname werden entweder vom Provider vorgegeben oder können bei einem isolierten Netz relativ frei gewählt werden. Da ein isoliertes Netz – fast – immer nach absehbarer Zeit Anschluss an das Internet findet, ist die Beachtung der Verbote und Regeln für ein an das Internet angeschlossenes, lokales Netz sinnvoll.

- Die Top-Level-Domain darf nicht *local* sein.
- Wenn der Provider keine Top-Level-Domain vergibt, ist eine nicht vergebene Top-Level-Domain zu wählen, wie *lcl* oder *loc*
- Der Name des lokalen Netzes darf nur dann der gleiche wie der des Providers sein, wenn dieser es ausdrücklich zulässt.
- Wenn der Provider Top-Level-Domain und Domain vergibt, wird der Computer-Name auch vergeben. Wenn keine dynamische Namensvergabe erfolgt, wird nicht nur der Name, sondern auch die Adresse fest vergeben.

Ansonsten kann der Name frei gewählt werden.

Versionsnummern

Unter Linux-Programmierern hat sich die Vergabe von dreiteiligen Versionsnummern

x.y.z

etabliert: Dabei steht

1. *x* für die Bezeichnung der Softwarearchitektur
2. *y* für die Hauptversion
3. *z* für die Nebenversion

Unterschiedliche Softwarearchitekturen bedingen völlig unterschiedliche Funktionen, Konfiguration und Bedienung. Anwendungen, die sich in der Softwarearchitektur unterscheiden, können nur nach umfangreicher Anpassung der Arbeitsumgebung ausgetauscht werden. Im Allgemeinen werden Hilfsprogramme zur Verfügung gestellt, die eine Migration entlang aufsteigender Softwarearchitekturversionen erlauben. Das Übertragen einer Anwendung von einer höheren Architekturversion zur einer niedrigeren bedeutet in den allermeisten Fällen eine Neuprogrammierung.

Zwischen unterschiedlichen Hauptversionen sollten Konfigurations- und Anwendungsdaten hin- und hergetauscht werden können. Gegebenenfalls müssen Programme neu übersetzt werden, Dateien sind gewöhnlich austauschbar und kompatibel, zumindest in aufsteigender Reihenfolge.

Bei unterschiedlichen Nebenversionen sollten keinerlei besondere Vorkehrungen getroffen werden müssen. In den meisten Fällen bedeuten höhere Nebenversionen Fehlerbehebungen gegenüber der Vorversion. Mit höheren Versionsnummern eingeführte zusätzliche Eigenschaften sollten keine Schwierigkeiten bereiten, zumindest der Austausch von niedriger zu höherer Versionsnummer darf keinen Aufwand verlangen.

Grundsätzlich muss vor einem Versionswechsel immer die versionspezifische Dokumentation (Release Notes) gelesen werden, da hier gelegentlich doch notwendige Maßnahmen beschrieben werden.

Distributionen

Bei der Vergabe von Versionsnummern für Distributionen sind zweiteilige Nummern üblich. Die erste Stelle verdeutlicht meist tiefer gehende Änderungen wie Wechsel der Hauptversion des Kernels oder Umstellung der Installations- und Konfigurationsmechanismen.

Die zweite Stelle wird für Softwarepflege und -aktualisierungen verwendet.

Die Verwendung von Versionsnummern unterliegt bei kommerziellen Distributoren gelegentlich Marketing-Interessen, größere Versionsprünge werden dabei aus Wettbewerbsgründen vorgenommen.

Kernel

Linux Torvalds hat für den Kernel die Vergabe von dreiteiligen Versionsnummern

$x.y.z$

eingeführt: Dabei steht

1. x für die Bezeichnung der Kernelarchitektur
2. y für die Hauptversion
3. z für die Nebenversion

Unterschiedliche Kernelarchitekturen bedingen eine Umstellung von Compiler, Bibliotheken und oft auch Programmierweise. Bei einem Wechsel müssen die gesamten Distributionen überarbeitet werden. Versionswechsel finden höchstens alle drei bis fünf Jahre statt.

Für den Kernel werden gerade Hauptversionsnummern für so genannte stabile Entwicklungslinien vergeben, ungerade für Test- und Entwicklungslinien. Die Nebenversionen einer Entwicklungslinie steigen meist sehr schnell an, bis die getestete Version zu einer Erhöhung der stabilen Entwicklungslinie um eins führt.

Kernel mit höheren Nebenversionen unterstützen fast immer eine wesentlich höhere Hardwarepalette als die jeweiligen Vorversionen.

Das Copyright von Linux

Der Linux-Kernel unterliegt der *GNU Public License (GPL)*, wie auch die allermeiste Software. Sie ist im Anhang zu finden.

Open Source und die philosophischen Ideen hinter Linux

Die Ideen, die hinter der Entwicklung von Linux stecken, wurden von *Eric S. Raymond* formuliert³⁰. Eine Übersetzung ist im Anhang zu finden.

Hardware-Anforderungen

Die unterstützte, wie die benötigte Hardware ist auf den Internetseiten der Provider³¹ und in den Newsgroups zu finden.

³⁰ <http://www.catb.org/~esr/writings/cathedral-bazaar/>

³¹ <http://bugzilla.redhat.com/hwcert/> und http://cdb.novell.com/?LANG=de_DE news://comp.os.linux.hardware/