

Übersetzung: Dr. phil. Doris Martin, contec d. & c. martin, Diedorf-Anhausen

Titel der Originalausgabe: „Just for FUN - The Story of an Accidental Revolutionary" © 2001 by Linus Torvalds and David Diamond. Published by HarperCollins Publishers, Inc., 10 East 53rd Street, New York, NY 10022

Der Abschnitt „Ist die Linuxrevolution vorbei?" von Scott Berinato wurde mit Genehmigung von ZD Net (26. August 1999) abgedruckt. © 1999 by ZD Net Inc.

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Ein Titeldatensatz für diese Publikation ist bei Der Deutschen Bibliothek erhältlich.

Dieses Werk ist urheberrechtlich geschützt. Alle Rechte, auch die der Übersetzung, des Nachdruckes und der Vervielfältigung des Buches, oder Teilen daraus, vorbehalten. Kein Teil des Werkes darf ohne schriftliche Genehmigung des Verlages in irgendeiner Form (Fotokopie, Mikrofilm oder ein anderes Verfahren), auch nicht für Zwecke der Unterrichtsgestaltung, reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

© 2001 Carl Hanser Verlag München Wien Lektorat:
Margarete Metzger Herstellung: Irene Weilhart
Umschlaggestaltung: Zentralbüro für Gestaltung, Augsburg
Gesamtherstellung: Kösel, Kempten Printed in Germany

ISBN 3-446-21684-7

Für Tove und Patricia, Daniela und Celeste. Ich wollte immer von jungen Frauen umschwärmt sein, und ihr habt mir diesen Traum erfüllt. Für Tia und Kaley. Mann, kann ich mich glücklich schätzen. Vermutlich würde das nicht als Dank durchgehen, wenn wir uns nicht mit ein paar wichtigen Namen schmücken würden. Also dann: Wir danken unserem Lektor Adrian Zackheim, der sich all unseren Wünschen beugte; Erin Richnow, der Lektoratsassistentin bei HarperCollins, die das Projekt besser im Griff hatte als wir; unseren Agenten, Bill Gladstone von Waterside Productions und Kris Dahl von ICM, die unsere Honorare nicht schneller hätten überweisen können; Sara Torvalds, die über das beste Gedächtnis auf der ganzen Fennoskandia-Halbinsel verfügt - und in drei Sprachen arbeitet -, und William und Ruth Diamond, die das Originalmanuskript lasen und immer wieder beteuerten: „Nein, es ist wirklich gut.“

„Um eines allerdings machte ich mir Sorgen, als er größer wurde: Wie um alles in der Welt sollte er auf diese Weise jemals nette Mädchen kennen lernen?“

Anna Torvalds

Vorwort

Normalerweise bittet man jemanden, der berühmter ist als der Autor, das Vorwort zu einem Buch zu schreiben, gemäß der Theorie, dass damit die Bedeutung des Autors betont wird. Nun, der Papst hatte keine Zeit, und Bill Gates ... doch lassen wir das. Ignoriert man dann noch ein paar Schauspieler, die Linux wahrscheinlich nicht von einem Waschmittel unterscheiden können, dann endet die Liste schon sehr bald.

Die Faszination des klassischen American Dream, des „David gegen Goliath“, der Revolution von unten. Alles Elemente, die natürlich das Interesse auf die Person dahinter lenken. Wie kann es sein, dass ein junger finnischer Student eine der wichtigsten Industrien unserer Zeit revolutioniert. Wie kommt es, dass die mächtigsten Konzerne der Welt sich mit den Ideen eines eher unscheinbaren jungen Mannes auseinan-

dersetzen, entweder, weil sie darin die Zukunft der Branche zu entdecken glauben, oder, weil hier die schlimmsten Albträume eines Endes herrlicher monopolistischer Zeiten auftauchen. Warum existiert in einer gar nicht mal kleinen Gruppe der Gesellschaft ein geradezu atemberaubender Starkult um jemanden, der eigentlich nur seine Ruhe will, um ungestört Billard zu spielen oder eben am Computer zu sitzen und seinem Hobby nachzugehen? Einem dunkelblonden, eher schwächlichen jungen Mann mit inzwischen recht deutlichem Bauchansatz und bestenfalls als „interessant“ zu bezeichnendem Modegeschmack und Stilbewusstsein.

Es ist die Kombination der Person Linus Torvalds, und der Bewegung namens Linux, oder allgemeiner, Open Source. Linus hat Open Source nicht erfunden, aber er hat mit Linux den wesentlichen Grundstein dafür gelegt, dass Open Source Erfolg haben konnte. Er hat seine

Ideen mit anfangs ein paar Dutzend, dann Hunderten, Tausenden und heute in der Tat Millionen Menschen geteilt. Und das, ohne dafür Geld zu fordern, sondern basierend auf der Idee, dass alle gemeinsam eben bessere Software schreiben können. Und das schneller und effizienter als jede Firma das gemäß **traditioneller Ideen der Softwareentwicklung könnte.**

Auf den ersten Blick mag das unsinnig erscheinen. Warum etwas umsonst weitergeben, wofür man doch Geld verlangen könnte. **Aber der wesentliche Punkt ist auch gar nicht das „umsonst“, sondern das „frei“ (was unglücklicherweise beides im Englischen mit „free“ bezeichnet werden kann).** Diese Freiheit, die Sourcen für Linux verwenden und modifizieren zu können ist der **Grundstein des Phänomens Linux.** Hierin begründet sich gerade die Möglichkeit, dass Linux sich so schnell verbreitet und weiterentwickelt. Und hierin liegt die Motivation so vieler hochbegabter Entwickler, sich aktiv in das Projekt einzubringen.

Und Linus selber, der trotz all des Wirbels ein erfrischend normaler und unkomplizierter Mensch geblieben ist, er fasziniert wahrscheinlich gerade deswegen. Man sieht ihn fast nie Vorträge halten, wenn er verreist, ist er nicht von Bodyguards umgeben, sondern meist von seiner Frau Tove und seinen drei kleinen Töchtern. Wenn er an Gesprächen teilnimmt, dann dominiert er selten, sondern überrascht vielmehr durch seine eher ruhig vorgetragenen, durchdachten und gelegentlich kontroversen Beiträge. Die Entwicklergemeinde respektiert ihn und schaut auf zu ihm, gerade weil er sich nicht im Rampenlicht sonnt, sondern lieber Taten für sich sprechen lässt. Und weil er sich konsequent weigert, sich vor den Marketing-Karren der Firmen im Linux-Umfeld spannen zu lassen.

Ein komischer Heiliger? Nein, ein Heiliger ist Linus sicher nicht. Er ist ein sympathischer Idealist, der sich einen ausgeprägten Pragmatismus und einen klaren Sinn für das Mögliche bewahrt hat. Der durchaus mal egoistisch, verspielt und verschwenderisch ist, aber bei all dem doch nie den Boden unter den Füßen zu verlieren scheint.

Das Buch ist kein Pamphlet, keine dogmatische Beschreibung von Theorien oder Ideologien. Auch kein durchgestylt-stromlinienförmiges Machwerk der Selbstbeweihräucherung. Kein typisches ManagerBuch, wie wir es in **den letzten Jahren immer wieder mal erleben und erlesen mussten.** Vielmehr ist es eine sehr persönliche und offene

Sammlung von Versatzstücken, schildert Linus' eigene Perspektive zu all dem Wirbel, bringt Anekdoten und Trivialitäten aus seinem Leben. Zusammen mit Beobachtungen von David Diamond liefert es einen unterhaltsamen, interessanten und sehr lesenswerten Einblick in die Ursprünge und Hintergründe der laufenden Revolution der IT Industrie. Und verrät nebenbei viel über die Hauptperson Linus Torvalds selbst, den Menschen hinter all dem.

Nürnberg, im Mai 2001

Dirk Hohndel

Inhalt

<i>Rückblick auf eine Revolution</i>	XV
<i>Dank</i>	
<i>Der Sinn des Lebens, 1</i> .	1
<i>Geburt eines Computerfreaks</i>	2
<i>Geburt eines Betriebssystems</i>	45
<i>Ballkönig</i>	135
Geistige Eigentum	219
Nieder mit der Gängelei	230
Die Karussellfahrt, die vor uns liegt . .	236
Warum Open Source einen Sinn ergibt	241
Ruhm und Reichtum	251
Der Sinn des Lebens, 2	258
	267

<i>Register</i>	
-----------------------	--

Rückblick auf eine Revolution

In der Euphorie der letzten Jahre des 20. Jahrhunderts ereignete sich eine von vielen Revolutionen. Praktisch über Nacht erregte das Betriebssystem Linux die Aufmerksamkeit der Welt. Mit einem explosiven Knall hatte es sich aus dem Jugendzimmer seines Schöpfers, Linus Torvalds, befreit und avancierte zum Kultobjekt einer besessenen Fangemeinde halbmilitanter Freaks. Plötzlich infiltrierte es die wirtschaftlichen Machtzentren, die den Planeten regierten. Was als Party eines Einzelnen begann, begeisterte binnen kurzem Millionen von Benutzern auf jedem Kontinent einschließlich der Antarktis und des Weltraums, wenn man die Außenposten der NASA mitrechnet. Das Betriebssystem steuerte nicht nur die Mehrzahl der Server, die den gesamten Inhalt des World Wide Web verteilten, sein einzigartiger Entwicklungsansatz - ein fein gewebtes Netz aus hunderttausenden von unentgeltlich arbeitenden Programmierern - war auch zum größten Gemeinschaftsprojekt in der Geschichte der Menschheit geworden. Die dahinter stehende Open-Source-Philosophie war einfach: Informationen, in diesem Fall der Quellcode oder die Grundbefehle hinter dem Betriebssystem, sollten *free and freely*, ungehindert und umsonst, jedem zugänglich sein, der Weiterentwicklungen daran vornehmen wollte. Im Gegenzug sollten die Weiterentwicklungen ebenfalls der Allgemeinheit uneingeschränkt zur Verfügung gestellt werden. Dieses Konzept hatte vorher bereits über Jahrhunderte hinweg die wissenschaftliche Erkenntnis befördert. Nun hielt es Einzug in Unternehmenssphären, und sein Potenzial als Rahmen für Höchstleistungen in jedem Bereich - als Strategie für einen Gerichtsprozess, eine Oper - wurde erkennbar.

Ein paar Leute erhaschten einen Blick in die Zukunft, und was sie sahen, gefiel ihnen nicht. Linus' rundes, bebrilltes Gesicht wurde zu

einer beliebten Zielscheibe auf den Dartboards der Microsoft Corporation, die sich erstmals mit einer echten Bedrohung durch einen Konkurrenten konfrontiert sah. Weitaus häufiger aber wollten die Leute mehr über den Jungen wissen, der das Ganze begonnen oder zumindest erdacht hatte, und praktisch das Oberhaupt von all dem war. Allerdings wollte dieser mit dem zunehmenden Erfolg von Linux und Open Source immer weniger darüber reden. Unser Zufallsrevolutionär hatte sich Linux ausgedacht, weil ihm das Herumspielen an einem Computer Spaß machte. (Und weil die Alternativen nicht besonders attraktiv waren.) Und so bot Linus, als man ihn mit dem Argument, seine Millionen von Anhängern wollten ihn wenigstens einmal live zu *sehen* bekommen, als Redner für eine wichtige Veranstaltung gewinnen wollte, freundlich an, stattdessen lieber an einem *Dunk-Tank* teilzunehmen.' Das sei unterhaltsamer, erklärte er. Und außerdem eine Möglichkeit, an Geld zu kommen. Die Veranstalter lehnten dankend ab. Unter einer Revolution verstanden sie etwas anderes. Revolutionäre werden nicht geboren. Revolutionen sind nicht planbar. Revolutionen sind nicht steuerbar.

Revolutionen *geschehen* einfach.

David Diamond

Anmerkung der Übersetzerin: Der Dunk-Tank ist in den USA eine Attraktion auf Jahrmärkten und in Fernseh-Shows: Mehr oder weniger prominente Personen nehmen auf einem Podest Platz, unter dem sich ein Wasserbecken befindet. Nun versuchen Leute, die dafür oft Geld bezahlt haben, durch gezielte Würfe mit einem Ball einen Mechanismus in Gang zu setzen, der zur Folge hat, dass das „Opfer“ im Wasserbecken landet.

```
X-Authentication-Warning: penguin.transmeta.com: torvalds
owned process doing -bs Date: Mon, 18 Oct 1999 14:12:27 -0700
(PDT) From: Linus Torvalds <torvalds@transmeta.com> To: David
Diamond <ddiamond@well.com> Subject: Ho humm.. MIME-Version:
1.0 Hoffentlich stimmt die Mail-Adresse noch. Ich habe fest-
gestellt, dass ich überhaupt keine Informationen habe, wo ich
dich erreichen kann, wahrscheinlich weil ich deine
Visitenkarte mit all den anderen weggeworfen habe, und weil
du mich viel öfter angerufen als angemailt hast. Ich habe am
Wochenende gründlich nachgedacht, und ich glaube, wenn du
weiterhin interessiert bist, gefällt mir der Gedanke immer
besser. Einigen wir uns doch auf einen Deal: Wenn du denkst,
dass wir ein Spaß-Buch machen können, und wenn du - was viel
wichtiger ist - glaubst, dass uns das Spaß machen wird,
sollten wir es anpacken. Du könntest mich (mit Familie) zum
Campen und (ohne Familie) zum Fallschirmspringen
mitschleppen. Sachen, die ich sonst nie tun würde, bloß weil
ich mir einbilde, zu viel zu tun zu haben. Liefere mir eine
Entschuldigung, all die Dinge zu tun, die ich in den letzten
drei Jahren versäumt habe, obwohl die Gelegenheit dazu prak-
tisch vor der Haustür liegt ... Auch wenn ich ein Buch über
mich vielleicht nicht lesen würde, wenn es fertig ist, hätte
ich auf diese Weise wenigstens meinen Spaß daran. Linus
```

...und manchmal lässt man Revolutionären einfach keine andere Wahl.

Linus Torvalds

Dank

Wir möchten den folgenden Einrichtungen für ihre Rolle bei der Entstehung dieses Buches danken - oder jedenfalls für den Spaß, den wir durch sie hatten. (Keine von ihnen hat uns dafür Geld gezahlt. Was eine verdammte Schande ist.)

FM 107.7 the Bone. Classic Rock That Rocks; Zelda's Restaurant, Capitola; Kiva Retreat House, Santa Cruz; Hagashi West Restaurant, Palo Alto; Malibu Grand Prix, Redwood Shores; Bodega Bay Lodge, Bodega Bay; Saturn Cafe, Santa Cruz; Cafe Marmelade, Ross; Half Moon Bay Boardshop, Half Moon Bay; Santa Cruz Billards, Santa Cruz; Cafe Reyes, Point Reyes Station; California Sushi and Grill, San Jose; Santa Clara Golf and Tennis Club, Santa Clara; Ideal Bar and Grill, Santa Cruz; Silver Peso Bar („Where Janis Played"), Larkspur; Rosie McCann's Irish Pub and Restaurant, Santa Cruz; Mayflower Inn, San Rafael; Grover Hot Springs State Park, Markleeville; Left Bank Restaurant, Larkspur; Potrero Brewing Company, San Francisco; The Rice Table, San Rafael; Ross Valley Swim and Tennis Club, Kentfield; Fallen Leaf Lake Marina, Fallen Leaf Lake; Peet's Coffee and Tea, Greenbrae; Hawthorne Lane Restaurant, San Francisco; Indian Springs Resort, Calistoga; Samurai Sushi, Sausalito; Blowfish Sushi, San Francisco; Paramount's Great America, Santa Clara; Robata Grill Sushi, Mill Valley; Buckeye Roadhouse, Mill Valley; Barnes and Noble, San Jose; Sushi Ran, Sausalito; 23 Ross Common, Ross; KFOG-104.5 FM; Rutherford Grill, Rutherford; In-N-Out Burgen Santa Rosa; Seto Sushi, Sunnyvale.

Der Sinn des Lebens, 1 (Sex, Krieg und Linux)

Schauplatz: Dieses Buch nimmt seinen Anfang in einem alten schwarzen Ford Expedition auf der Interstate 5 in südlicher Richtung, irgendwo im Central Valley Kaliforniens. Ein Störenfried begleitet Linus und Tove Torvalds und ihre kleinen Töchter Patricia und Daniela auf ihrer 351 Meilen langen Fahrt nach Los Angeles, wo sie den Zoo und IKEA besuchen wollen.

David: Also es gibt da eine grundlegende Frage, über die wir nachdenken müssen, und die ziemlich wichtig ist. Was willst du mit diesem Buch eigentlich rüberbringen?

Linus: Na ja, ich will den Sinn des Lebens erklären.

Tove: Linus, hast du daran gedacht zu tanken?

L: Ich habe eine Theorie über den Sinn des Lebens. Die können wir den Leuten im ersten Kapitel erklären. Das weckt ihr Interesse. Wenn sie dann angebissen haben und das Buch kaufen, können wir den Rest einfach mit irgendwelchem Unsinn füllen. Verstehe, das klingt nach einem Plan. Jemand hat mir mal erzählt, dass zwei Fragen seit Anbeginn der Menschheit ungelöst geblieben sind: Erstens: „Was ist der Sinn des Lebens?“ und zweitens: „Was mach ich mit all dem Kleingeld, das sich am Abend in meinen Taschen angesammelt hat? “

D:

L: Die Antwort auf die erste Frage habe ich.

D: Und die wäre?

L: *Eigentlich ist sie kurz und einfach. Sie wird deinem Leben zwar keinen Sinn geben, aber du weißt dann wenigstens, womit du rechnen musst. Drei Sachen sind im Leben wirklich wichtig. Sie sind die Antriebsfaktoren für alles in deinem Leben - für alles was du oder ein anderes Lebewesen tust. Das erste Motiv ist das Überleben, das zweite die Gesellschaftsordnung und das dritte Vergnügen und Unterhaltung. Alles im Leben geschieht in dieser Reihenfolge. Und nach dem Vergnügen kommt nichts anderes mehr. Das heißt gewissermaßen, der Sinn des Lebens besteht darin, diese dritte Stufe zu erreichen. Sobald du die dritte Stufe erreicht hast, hast du's geschafft. Aber erst musst du die anderen Stufen durchlaufen. Das musst du mir mal genauer erklären.*

D:

Patricia: *Papi, können wir anhalten und ein Schokoladeneis kaufen? Ich würde jetzt gern ein Schokoladeneis essen.*

T *Nein, Schatz. Du musst noch warten. Du darfst ein Eis haben, wenn wir anhalten, um aufs Klo zu gehen.*

L: *Ich werde dir ein paar Beispiele nennen, damit die Sache klarer wird. Nehmen wir Sex, das ist am nahe liegendsten. Sex diente erst mal dem Überleben, aber dann wurde etwas Gesellschaftliches daraus. Deshalb heiratet man. Und dann wird er zum Vergnügen.*

P: *Dann muss ich jetzt aufs Klo.*

D: *Inwiefern zum Vergnügen?*

L: *Ok, ich rede mit der falschen Person. Vielleicht sollte ich lieber*

Nein, bleib beim Thema.

D: *Es gibt da auch noch eine andere Ebene*

L:

D(zu sich selbst): *Oh, ein Vergnügen, es zu tun, nicht ein Vergnügen, dabei zuzusehen. Okay, ich hab's verstanden.*

L: *.... eine andere Ebene, wenn du die Illusion von Sex in einem biologischen Sinn betrachtest. Wie entstand Sex überhaupt? Aus Überlebensgründen. Zuerst ging es nicht ums Vergnügen, nur darum, sich zu vereinigen. Okay, lassen wir das. Gerede über Sex.*

D: *Nein, nein. Ich denke, daraus lässt sich ein ganzes Kapitel machen.*

L: *Nehmen wir stattdessen Krieg. Wenn irgendein Riesenkerl dir den Weg zum Wasserloch versperrt, geht es offensichtlich erst mal ums Überleben. Als Nächstes musst du mit dem Typen um eine Frau kämpfen. Und dann wird Krieg zu einer Frage der Gesellschaftsordnung. So war es schon lange vor dem Mittelalter. Krieg als Mittel, eine Gesellschaftsordnung herzustellen.*

D: *Genau. Und auch ein Mittel, sich selbst als Teil der Gesellschaftsordnung zu etablieren. Niemand interessiert sich für die Gesellschaftsordnung an sich. Was uns interessiert, ist unser eigener Platz in dieser Ordnung. Menschen leben in Rangordnungen wie Hühner in Hackordnungen. Da gibt es keinen großen Unterschied.*

Und jetzt ist Krieg zur Unterhaltung da?

Ja, genau.

D: *Vielleicht für Leute, die ihn sich im Fernsehen anschauen.*

Für die mag Krieg ja unterhaltsam sein.

L: *Computerspiele. Kriegsspiele. CNN. Die Auslöser eines Krieges*

D: *können oft unterhaltsam sein. Aber es ist auch unterhaltsam, sich Kämpfe und Schlachten anzuschauen. Und der Grund für Sex ist oft unterhaltsam. Klar, die Überlebensfrage ist immer noch ein*

L: *Punkt, vor allem wenn man katholisch ist, stimmt's? Aber selbst wenn du katholisch bist, denkst du vermutlich manchmal auch an den vergnüglichen Teil. Es muss also nicht reinweg ums Vergnügen gehen. Bei allem was wir tun, kann teilweise das Überleben eine Rolle spielen, teilweise die Gesellschaftsordnung und der Rest sind Vergnügen und Unterhaltung. Okay, schau dir doch mal die Technik an. Die Technik sollte zunächst einmal das Überleben erleichtern. Und Überleben bedeutet nicht nur zu überleben, sondern besser zu überleben. Und dann hast du plötzlich eine Windmühle, die Wasser aus dem Brunnen zieht ...*

Oder Feuer.

D:

- L: Genau. Bei all diesen Dingen geht es immer noch ums Überleben; die Gesellschaftsordnung und der Unterhaltungsfaktor spielen noch keine Rolle.
- D: Und wie hat die Technik es dann geschafft, zu einer Frage der Gesellschaftsordnung zu werden?
- L: Na ja, ein Großteil der Industrialisierung hatte nur damit zu tun, zu überleben oder besser zu überleben. Also zum Beispiel schnellere und schönere Autos zu bauen. Aber reden wir mal über Technik in einem gesellschaftlichen Sinn. Hier kommt das Telefon ins Spiel. Und bis zu einem gewissen Grad auch das Fernsehen. Viele der ersten Fernsehsendungen dienten im Grunde der Indoktrination. Beim Radio war es genauso. Deshalb investierten Länder oft als Erstes in den Rundfunk, wegen seiner gesellschaftspolitischen Seite.
- D: Um eine Gesellschaftsordnung aufzubauen und aufrechtzuerhalten ...
- L: Genau, aber irgendwann wurde mehr daraus. Heute sehen wir vor allem zum Vergnügen fern, das ist offensichtlich. Und mittlerweile bist du überall von Mobiltelefonen umgeben. Das ist ein gesellschaftliches Phänomen. Aber auch der Unterhaltungsfaktor spielt zunehmend eine Rolle.
- D: Wo liegt dann also die Zukunft der Technik? Mittlerweile haben wir die Stufe des Überlebens hinter uns gelassen und befinden uns auf der gesellschaftlichen Stufe, richtig?
- L: Richtig. Bisher leistete die Technik nichts anderes, als uns das Leben zu erleichtern. Es ging immer nur darum, schneller zu reisen, billiger zu produzieren, besser zu wohnen, was auch immer. Was ist jetzt an der Informationstechnologie so anders? Was kommt nach der Tatsache, dass jeder am Netz hängt? Was bleibt noch zu tun? Klar, die Verbindungen können noch besser werden, aber das ist nichts grundsätzlich Neues. Wohin führt uns also die Technik? Meiner Meinung nach ist der nächste große Schritt die Unterhaltung.
- D: Alles wird irgendwann zum Vergnügen.
- L: Aber das erklärt bis zu einem gewissen Grad auch den Erfolg von Linux. Denk an die drei Antriebsfaktoren. Der erste ist das Überleben, das Leute mit Computern als selbstverständ-

lich betrachten. Ehrlich gesagt, wenn du einen Computer hast, hast du dir dein Essen und dergleichen schon gekauft. Der zweite ist die Gesellschaftsordnung und die gesellschaftliche Seite ist für Freaks, die in ihren Minibüros sitzen, mit Sicherheit ein Motivationsanreiz.

- D: Du hast bei der Comdex etwas wirklich Tiefgründiges gesagt, etwas in der Richtung, die Entwicklung von Linux sei ein globaler Mannschaftssport. Hey, Mann, das ist dein Werk. Linux ist ein fabelhaftes Beispiel, warum Leute so begeistert von Mannschaftssportarten sind und darauf brennen, Teil eines Teams zu sein.
- L: Yeah, wenn du den ganzen Tag vorm Computer sitzt, brauchst du vermutlich das Gefühl, Teil von etwas zu sein. Egal, was. Es hat etwas mit sozialen Kontakten zu tun, wie jeder andere Mannschaftssport. Denk bloß an die Spieler einer Football-Mannschaft, besonders in der High School. Der gesellschaftliche Aspekt von Linux ist wirklich unglaublich wichtig. Aber Linux ist auch Unterhaltung, und zwar die Art von Unterhaltung, die du mit Geld nur sehr schwer kaufen kannst. Geld ist ein mächtiger Motivationsanreiz, wenn du dich auf der Stufe des bloßen Überlebens befindest, weil Überleben sich einfach kaufen lässt. Das meiste, was du dafür brauchst, bekommst du bequem im Supermarkt. Aber wenn du auf der Stufe der Unterhaltung angelangt bist, ist Geld plötzlich ... Du meinst, dann ist Geld nutzlos?
- D: Nein, nicht nutzlos, denn du kannst dir damit natürlich Filme, schnelle Autos oder Urlaube kaufen. Du kannst eine Menge Dinge kaufen, die dein Leben bereichern. Linus, wir müssen Daniela wickeln. Und Patricia muss aufs Klo. Und ich habe Lust auf einen Cappuccino. Denkst du, wir finden hier irgendwo einen Starbucks? Wo sind wir denn überhaupt?
- T:

D (schaut hoch): Dem Geruch nach zu schließen, müssen wir in der Nähe von King City sein.

- L: Heute spielt sich das Ganze in einer anderen Größenordnung ab. Es betrifft nicht nur einzelne Menschen, sondern

das Leben an sich. Es ist wie das Gesetz der Entropie. In diesem Entropiegesetz des Lebens bewegt sich alles weg vom Überleben hin zur Unterhaltung. Das schließt jedoch lokale Rückschritte, wie wir sie ständig erleben, nicht aus. Manchmal laufen die Dinge einfach aus dem Ruder.

D: Aber als System bewegt sich alles in die gleiche Richtung. Alles bewegt sich in die gleiche Richtung, aber nicht zur gleichen Zeit. Sex hat das Unterhaltungsstadium erreicht, Krieg steht kurz davor, Technik hat es im Prinzip geschafft. Neue Sachen dienen zuerst einmal nur dem Überleben. Zum Beispiel die Raumfahrt: Sie wird irgendwann eine Überlebensfrage sein, danach zu einer Frage der Gesellschaftsordnung werden und schließlich der Unterhaltung dienen. Oder schau dir den Kult um die Kultur an. Ich meine, auch hier haben wir wieder das gleiche Muster. Kultur und Zivilisation dienen zunächst dem Überleben. Menschen tun sich zusammen, um besser überleben zu können und bauen ihre Gesellschaftsstruktur auf. Dann, irgendwann, existiert Kultur ausschließlich um der Unterhaltung willen. Okay, gut, nicht ausschließlich. Und sie braucht deswegen keine schlechte Unterhaltung zu sein. Es ist bekannt, dass die Griechen eine sehr starke Gesellschaftsordnung hatten und sich dabei bestens unterhielten. Sie hatten die besten Philosophen ihrer Zeit. Okay, wie hängt das alles mit dem Sinn des Lebens zusammen?

D: Tut es eigentlich nicht ... Es bedeutet nur, dass ... Das ist irgendwie noch ein Problem.

L: Das ist das kleine fehlende Glied in der Kette, über das du noch nachdenken musst.

Mama, sieh mal, da sind Kühe.

D: Also, wenn du weißt, dass es im Leben auf diese Art der Weiterentwicklung ankommt, dann ist es offensichtlich dein Ziel im

P: Leben, diese Weiterentwicklung zu vollziehen. Und die

L: Weiterentwicklung ist nicht nur eine einzelne Weiterentwicklung.

Alles, was du tust, ist ein Teil vieler verschiedener Weiterentwicklungen. Die Frage könnte genauso gut heißen: „Was kann ich tun, um die Gesellschaft besser zu machen?“

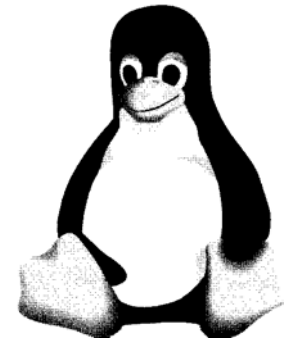
Du weißt, dass du ein Teil der Gesellschaft bist. Du weißt, dass die Gesellschaft sich in diese Richtung bewegt. Und dabei kannst du ihr behilflich sein.

T (hält sich die Nase zu): Es riecht schrecklich hier.

L: All das läuft also darauf hinaus, dass wir letztlich alle hier sind, um Spaß zu haben. Eigentlich könnten wir uns einfach entspannt zurücklehnen und die Fahrt genießen

D: Just for fun? Einfach so zum Spaß?

Geburt eines Computerfreaks



Ich war ein hässliches Kind.

Was soll ich sagen? Ich hoffe, eines Tages wird jemand einen Film über Linux drehen, und bestimmt werden sie die Hauptrolle mit jemandem besetzen, der aussieht wie Tom Cruise, aber in der Nicht-Hollywood-Version liegt der Fall meistens anders.

Verstehen Sie mich nicht falsch. Es ist nicht so, als würde ich wie der Glöckner von Notre Dame aussehen. Aber stellen Sie sich große Schneidezähne vor, so dass jeder, der ein Jugendphoto von mir sieht, unwillkürlich an einen Biber denken muss. Stellen Sie sich des Weiteren einen nicht vorhandenen Kleidergeschmack gepaart mit der traditionell übergroßen Torvalds-Nase vor, und das Bild in Ihrem Kopf kommt der Realität schon ziemlich nahe.

Meine Nase, sagt man mir manchmal, sei „stattlich“. Und es heißt -also, zumindest in unserer Familie heißt es das -, die Größe der Nase eines Mannes lasse auf die Größe anderer Dinge schließen. Aber sagen Sie das einem heranwachsenden Jungen, und er wird nicht viel darauf geben. Für ihn taugt diese Nase nur dazu, die Zähne zu überschatten. Das Bild mit den Profilen dreier Generationen von Torvalds-Männern ist allenfalls eine peinliche Erinnerung daran, mehr - ja, mehr Nase als Mann zu sein. Jedenfalls kam es mir damals so vor.

Nun brauchen Sie nur noch die Details zu ergänzen und Sie sind voll im Bilde. Braunes Haar (hier in den USA läuft es noch unter blond, aber in Skandinavien ist es schlicht „braun“), blaue Augen und eine leichte Kurzsichtigkeit, die das Tragen einer Brille ratsam erscheinen lässt. Und da eine Brille mit etwas Glück die Aufmerksamkeit von der Nase ablenkt, trage ich sie auch. Unentwegt.

Ach ja, den unsäglichen Geschmack in Kleidungsfragen habe ich schon erwähnt. Blau war meine Standardfarbe, und für gewöhnlich bedeutete das Jeans mit einem blauen Rollkragenpullover. Oder einem türkisfarbenen. Wie auch immer. Glücklicherweise wurden in unserer Familie nie viele Photos gemacht. So gibt es weniger belastendes Beweismaterial.

Ein paar Photos gibt es trotzdem. Auf einem davon bin ich ungefähr dreizehn Jahre alt und gemeinsam mit meiner sechzehn Monate jüngeren Schwester Sara zu sehen. Sie ist gut getroffen. Aber ich sehe aus wie eine Bohnenstange, ein magerer, blasser Teenie, der für die Photographin, vermutlich meine Mutter, posierte. Wahrscheinlich schoss sie dieses wunderbare Photo, als sie schon halb auf dem Weg in ihr Büro in der Finnish News Agency war, bei der sie als Übersetzerin arbeitete.

Da ich am Jahresende, am 28. Dezember, geboren bin, war ich so ziemlich der Jüngste in der Klasse. Und damit auch der Kleinste. Später spielt es keine Rolle mehr, wenn du ein halbes Jahr jünger bist als die meisten deiner Mitschüler. Aber in den ersten Schuljahren schon.

Und wissen Sie was? Erstaunlicherweise war das alles gar nicht so wichtig. Es spielte gar keine so große Rolle, dass ich ein bebrillter Zwerg mit dem Gesicht eines Bibers war, an den meisten Tagen ungepflegte Haare hatte (und an allen übrigen *extrem* ungepflegte) und mich unmöglich anzog. Weil ich so ein einnehmendes Wesen hatte.

Leider nein.

Nein, seien wir ehrlich, ich war ein Freak. Ein Nerd. Ein Geek. Praktisch von klein auf. Ich habe meine Brille nicht mit Klebeband zusammengehalten, aber ich hätte es ebenso gut tun können, denn alle anderen Merkmale waren vorhanden. Gut in Mathe, gut in Physik, null soziale Kompetenz. In einer Zeit, in der es noch nicht angesagt war, ein Nerd zu sein.

Wahrscheinlich hat fast jeder in seiner Schulzeit einen Typen wie mich gekannt. Der Junge, der der Beste in Mathe ist - nicht weil er wie verrückt lernt, sondern einfach so. In meiner Klasse war ich dieser Junge.

Aber lassen Sie mich die Geschichte weiter erzählen, bevor ich anfangen, Ihnen leid zu tun. Obwohl ich ein Nerd war und ein Zwerg, ging es mir ganz gut. Ich war nicht eben sportlich, aber ich war auch

bei dem zwei Teams versuchten, einander durch gezielte Treffer mit dem Ball zu dezimieren. Und auch wenn ich nie der absolute Spitzenspieler war, so gehörte ich doch meistens zu den ersten, die für eines der beiden Teams ausgewählt wurden.

Ich will damit sagen, dass ich zwar vom Auftreten her ein Nerd gewesen sein mag, insgesamt aber meine Schulzeit ziemlich genossen habe. Gute Noten, ohne viel dafür tun zu müssen. Nie wirklich tolle Noten, einfach weil ich *nichts* dafür tat. Und einen akzeptierten Platz in der Hackordnung. Niemand außer mir schien sich allzu sehr für meine Nase zu interessieren, wahrscheinlich weil die anderen, rückblickend betrachtet, genug mit ihren eigenen Problemen zu tun hatten.

Wenn ich mich so zurückerinnere, hatten auch die meisten anderen Kinder einen ziemlich schlechten Kleidergeschmack. Und wenn wir erwachsen werden, kümmern sich dann ohnehin andere um dieses heikle Problem. In meinem Fall sind das die Marketingexperten der Hightech-Unternehmen, die Leute, die die T-Shirts und Jacken auswählen, die man bei Tagungen von den Firmen geschenkt bekommt. Heute ziehe ich fast ausschließlich solche Werbegeschenke an, so dass

ich nie mehr Kleidung aussuchen muss. Und ich habe eine Frau, die für den Rest meiner Garderobe sorgt und Sachen wie Sandalen und Socken für mich auswählt. Deshalb brauche ich mich damit nicht mehr zu befassen.

Und an meine Nase habe ich mich gewöhnt. Zumindest im Augenblick bin ich mehr Mann als Nase.

II.

Es wird wahrscheinlich niemanden überraschen, dass es zu meinen frühesten und schönsten Erinnerungen gehört, mit dem alten elektronischen Tischrechner meines Großvaters zu spielen.

Er war der Vater meiner Mutter, Leo Waldemar Törnqvist, und Statistikprofessor an der Universität Helsinki. Ich erinnere mich, dass es mir irrsinnigen Spaß gemacht hat, den Sinus verschiedener Zufallszahlen zu berechnen. Nicht weil mich die Antwort so besonders interessiert hätte (wen würde das schon interessieren), sondern weil wir damals in einer anderen Zeit lebten, und Tischrechner einem die Antwort nicht einfach so lieferten. Sie *berechneten* sie. Und während sie das taten, blinkten sie wie verrückt, hauptsächlich um einen wissen zu lassen: „Ja, ich lebe noch, und ich brauche zehn Sekunden, um diese Berechnung durchzuführen, und in der Zwischenzeit blinke ich für dich, damit du siehst, wie viel ich arbeite.“

Das war faszinierend. Viel aufregender als ein moderner Rechner, der die simple Aufgabe, den einfachen Sinus einer Zahl zu berechnen, ohne mit der Wimper zu zucken erledigt. Bei diesen frühen Geräten wusste man, dass das, was sie taten, *schwierig* war. Sie ließen keinen Zweifel daran.

Ich weiß nicht mehr genau, wann ich das erste Mal einen Computer gesehen habe, aber ich muss wohl so um die elf gewesen sein. Wahrscheinlich war es 1981, als mein Großvater einen neuen Commodore VIC-20 kaufte. Da schon sein alter Tischrechner eine solch magische Anziehungskraft auf mich ausgeübt hatte, muss ich unheimlich aufgeregt gewesen sein und danach gelehzt haben, an den neuen Computer heranzukommen - aber irgendwie scheine ich mich nicht mehr daran zu erinnern. Ehrlich gesagt, ich weiß nicht

mich wirklich für Computer zu interessieren begann. Es fing langsam an und packte mich immer mehr.

Der VIC-20 war einer der ersten Computer von der Stange, die für den Hausgebrauch gedacht waren. Er musste nicht erst zusammengebaut werden. Man schloss ihn einfach an den Fernseher an und schaltete ihn ein, und das war's. Oben auf dem Bildschirm stand breit ein in Großbuchstaben geschriebenes „READY“ und ein fetter blinkender Cursor wartete darauf, dass man anfing.

Das Problem dabei war nur, dass man mit dem Ding nicht allzu viel anfangen konnte. Besonders in der ersten Zeit, als noch keine Infrastruktur für kommerzielle Programme entstanden war. Das Einzige, was man wirklich damit machen konnte, war BASIC-Programmierung. Und genau das tat mein Großvater.

Für meinen Großvater war dieses neue Spielzeug nicht nur ein Spielzeug, sondern auch eine grandiose Rechenmaschine. Der Computer konnte nicht nur den Sinus einer Zahl viel schneller berechnen als der alte elektronische Tischrechner, man konnte ihn obendrein dazu bringen, die Berechnung automatisch immer wieder durchzuführen. Außerdem konnte Großvater nun eine Menge der Dinge, die er an der Uni mit den großen Computern erledigt hatte, zu Hause erledigen.

Daran wollte er mich teilhaben lassen. Nebenbei versuchte er, mein Interesse für Mathematik zu wecken.

Also saß ich auf seinem Schoß, und er ließ mich seine Programme eingeben, die er sorgfältig auf Papier geschrieben hatte, weil es ihm schwer fiel, auf einer Tastatur zu tippen. Ich weiß nicht, wie viele andere elf- oder zwölfjährigen Jungen im Arbeitszimmer ihres Großvaters saßen und lernten, wie man arithmetische Ausdrücke vereinfacht und korrekt in einen Computer eingibt, aber ich erinnere mich **daran, dass** es bei mir so war. Ich weiß nicht mehr, worum es bei den Berechnungen ging, und ich glaube nicht, dass ich auch nur die leiseste Ahnung hatte, was ich da eigentlich tat. Auf jeden Fall aber saß ich mit Großvater am Computer und half ihm. Wahrscheinlich haben wir zu zweit viel länger gebraucht, als er allein gebraucht hätte, aber wer weiß das schon? Ich lernte, mit der Tastatur umzugehen, mit der mein Großvater nie richtig vertraut wurde. Ich beschäftigte mich nach der **Schule** damit, oder wann immer mich meine Mutter in der Wohnung meiner Großeltern absetzte.

Und ich begann, die Handbücher für den Computer zu lesen und die Beispielprogramme einzugeben. Es gab Beispiele für einfache Spiele, die man selbst programmieren konnte. Wenn man es richtig machte, hatte man am Schluss ein Männchen in primitiver Grafik, das über den Bildschirm lief. Und dann konnte man es verändern und das Männchen in verschiedenen Farben über den Bildschirm laufen lassen. Man konnte das einfach so tun.

Das ist ein unglaublich tolles Gefühl.

Und so fing ich an, meine eigenen Programme zu schreiben. Das erste Programm, das ich schrieb, war das erste Programm, mit dem jeder mal angefangen hat.

```
10 PRINT „HELLO“ 20  
GOTO 10
```

Das bewirkt genau das, was man erwartet. Es gibt HELLO auf dem Bildschirm aus. Bis in alle Ewigkeit. Oder zumindest, bis man es aus Langeweile unterbricht.

Aber es ist der erste Schritt. Manche Leute geben schon an diesem Punkt auf. Sie sehen darin nichts als eine einfältige Übung, denn warum sollte man eine Million Mal HELLO auf den Bildschirm ausgeben wollen? Trotzdem fingen die Handbücher zu den ersten Heimcomputern unweigerlich mit diesem Beispiel an.

Das Unglaubliche daran aber ist, dass man das Beispiel verändern kann. Meine Schwester behauptet, dass ich eine radikale zweite Version des Programms erstellte, die nicht schlicht HELLO ausgab, sondern stattdessen SARA IST DIE BESTE auf den Bildschirm schrieb, immer und immer wieder. Normalerweise war ich kein besonders liebevoller älterer Bruder. Wahrscheinlich hat diese Geste sie deshalb so nachhaltig beeindruckt.

Ich selbst kann mich nicht daran erinnern. Für gewöhnlich vergaß ich ein Programm, sobald ich es geschrieben hatte, und wandte mich dem nächsten zu.

III.

Lassen Sie mich von Finnland erzählen. Manchmal färbt sich der Himmel schon im Oktober schmuddelig-grau, und es sieht immer so aus, als würde es gleich regnen oder schneien. Jeden Morgen wacht man mit dieser düsteren Vorahnung auf. Der Regen ist eiskalt und schwemmt jede Erinnerung an den Sommer weg. Der Schnee hat die magische Eigenschaft, alles zu überstrahlen und mit einem Anflug von Optimismus zu überziehen. Das Problem dabei ist nur, dass der Optimismus ungefähr drei Tage lang anhält, während der Schnee Monat um klirrend kalten Monat liegen bleibt.

Im Januar bewegst du dich in einer Art Dämmerzustand, wenn du dich **nach draußen** wagst. Es ist die Jahreszeit, in der du dich in klamme, unförmige Klamotten zwängst und auf dem Weg zum Bus Gefahr **läufst** auszurutschen, wenn du die Abkürzung über den Oberschulsportplatz nimmst, auf dem eine Eishockeybahn angelegt ist. Auf Helsinkis Straßen musst du der einen oder anderen taumelnden älteren Dame ausweichen, die im September vermutlich noch eine ganz normale liebe Omi war, aber um 11 Uhr morgens an einem Dienstag im Januar nach einem Wodka-Frühstück im Zickzackkurs den Bürgersteig entlang wankt. Wer wollte es ihr verdenken? In ein paar Stunden wird es wieder dunkel sein, und es gibt nicht viel zu tun. Aber ich hatte einen Indoor-Sport, der mich durch den Winter brachte: Programmieren.

Morfar (das schwedische Wort für „Mutters Vater“) arbeitet häufig zu Hause, aber nicht immer. Er hat nichts dagegen, wenn du in seinem Zimmer sitzt, solange er weg ist. Du erbettelst das Geld für deinen ersten Computer. Alles ist in Englisch geschrieben, so dass du erst die Sprache decodieren musst. Es ist schwer, technische Literatur in einer Sprache zu verstehen, die du nicht sehr gut kannst. Du gibst dein Taschengeld für Computerzeitschriften aus. Eine davon enthält ein

Programm für Morsezeichen. Komischerweise ist das Programm nicht in der Sprache BASIC, sondern als Liste von Zahlen geschrieben, die sich per Hand in Maschinensprache übersetzen lassen - in die Nullen und Einsen, die der Computer versteht.

Und so entdeckst du, dass der Computer nicht wirklich BASIC spricht. Stattdessen arbeitet er mit einer viel einfacheren Sprache. Die Kinder in Helsinki spielen Eishockey und gehen mit ihren Eltern Langlaufen in den Wäldern. Du lernst, wie ein Computer wirklich arbeitet. Nicht ahnend, dass es Programme gibt, die für Menschen lesbare Zahlen in die Nullen und Einsen übersetzen, die der Computer versteht, fängst du einfach an, Programme in Zahlenform zu schreiben und per Hand umzuwandeln. Auf diese Weise programmierst du in Maschinensprache, und kannst dadurch Dinge tun, die du vorher nicht für möglich gehalten hättest. Du kannst die Fähigkeiten des Computers ausreizen. Du kontrollierst jedes einzelne kleine Detail. Du fängst an zu überlegen, wie du das Gleiche etwas schneller und mit weniger Speicherplatz tun kannst. Weil es zwischen dir und dem Computer keine Abstraktionsschicht gibt, kommst du ziemlich nahe an ihn heran. So ist es, wenn man mit einer Maschine intim ist.

Du bist zwölf, dreizehn, vierzehn, was auch immer. Andere Kids spielen draußen Fußball. Der Computer deines Großvaters ist interessanter. Er ist eine Welt für sich, in der die Logik regiert. In deiner Klasse haben vielleicht drei Leute einen Computer, und nur einer von ihnen nutzt ihn aus den gleichen Gründen wie du. Ihr trefft euch einmal die Woche. Es ist deine einzige soziale Aktivität, außer wenn einer deiner Computerfreunde gelegentlich bei dir übernachtet. Es ist dir egal. Die Sache macht einfach Spaß.

Das ist nach der Scheidung. Papa lebt in einem anderen Teil von Helsinki. Er denkt, sein Junge sollte mehr als ein Interesse haben. Deshalb meldet er dich fürs Basketball an, seinen Lieblingssport. Es ist eine Katastrophe. Du bist der Zwerg im Team. Nach eineinhalb Spielzeiten schleuderst du ihm mit hässlichen Worten an den Kopf, dass jetzt Schluss ist, dass Basketball *sein* Sport ist, nicht deiner. Dein neuer Stiefbruder, Leo, erweist sich als sportlicher. Und dann wird er auch noch Lutheraner, wie 86 Prozent der finnischen Bevölkerung. Da weiß Papa, der unerschütterliche Agnostiker, dass er als Vater versagt hat -vermutet hatte er es bereits ein paar Jahre früher, als Sara in die katholische Kirche eintrat.

Der Großvater mit dem Computer ist kein Kumpeltyp. Er bekommt langsam eine Glatze und ist leicht übergewichtig. Er ist der sprichwörtliche zerstreute Professor, und du kommst nur schwer an ihn ran. Er ist nicht gerade extrovertiert. Am besten stellt man ihn sich als einen Mathematiker vor, der nachdenkt und schweigend ins Leere starrt. Man wusste nie, worüber er gerade nachdachte. Komplexitätsanalyse? Frau Sammalkorpi am anderen Ende des Flurs? Ich bin genauso - berüchtigt dafür, mich auszuklinken. Wenn ich vor dem Computer sitze, reagiere ich auf jede Störung gereizt und verärgert. Tove kann ein Lied davon singen.

Meine lebhafteste Erinnerung an Morfar ist kein Computererlebnis, sondern hat mit seinem kleinen roten Ferienhaus zu tun. In Helsinki war es üblich, dass man ein kleines Ferienhaus hatte, das aus einem einzigen neun mal neun Meter großen Raum bestand. Die kleinen Häuser stehen auf einem handtuchgroßen Grundstück, und die Leute fahren dort hin, um in ihrem Garten zu werkeln. Typischerweise haben sie eine Wohnung in der Stadt und dazu diesen kleinen Garten, um Kartoffeln anzubauen, ein paar Apfelbäume zu pflegen oder Rosen zu züchten. Die meisten von ihnen sind schon etwas älter, denn die jüngere Generation hat aus beruflichen Gründen nicht die Zeit **dafür**. Unter diesen Leuten herrschte ein geradezu lächerlicher Wettbewerb um ihre gärtnerischen Erfolge. Dort pflanzte Morfar meinen Apfelbaum, einen kleinen Schössling. Vielleicht steht er immer noch da, es sei denn, er hat sich so prächtig entwickelt, dass ein neidischer Nachbar in einer kurzen Sommernacht auf das Grundstück schlich, um ihn abzusägen.

Vier Jahre, nachdem er mir die Welt der Computer eröffnet hat, erleidet Morfar einen Schlaganfall und bleibt einseitig gelähmt. Das ist für alle ein Schock. Er liegt ungefähr ein Jahr im Krankenhaus. Er ist einer der engsten Verwandten, die du hast, aber es berührt dich nicht besonders. Vielleicht ist es eine Schutzreaktion oder vielleicht sind Jungen in diesem Alter einfach so gefühllos. Er ist überhaupt nicht mehr der Mensch, der er mal war, und du besuchst ihn nicht gern. Du gehst **vielleicht alle** zwei Wochen hin. Deine Mutter besucht ihn öfter. Deine **Schwester**, die von klein auf die Rolle der Sozialarbeiterin der Familie übernahm, besucht ihn noch öfter.

Nach seinem Tod zieht sein Computer bei dir ein. Darüber gibt es eigentlich keine Diskussion.

Lassen Sie uns einen Moment Rückschau halten.

Finnland mag heute das angesagteste Land der Welt sein, aber vor Jahrhunderten war es kaum mehr als eine Zwischenstation für die Wikinger, als sie mit Konstantinopel „Handel“ trieben. Später, als die benachbarten Schweden Finnland befrieden wollten, entsandten sie den in England geborenen Bischof Henry, der im Jahre 1155 mit einem Missionsauftrag der katholischen Kirche kam. In ihrem Bekehrungseifer bemannten die Schweden die finnischen Festungen, um die Russen abzuwehren, und gewannen schließlich den Kampf um die Herrschaft gegen das Reich in unserem Osten. Um in den folgenden Jahrhunderten die Besiedelung der finnischen Kolonie voranzutreiben, bot man den Schweden Land und Steuervorteile an. Die Schweden schmissen den Laden bis 1714, als Russland ein siebenjähriges Gastspiel gab. Dann gewann Schweden seine Kolonie bis 1809 zurück, als Russland und Napoleon Finnland angriffen, das anschließend bis zur kommunistischen Revolution 1917 unter russischer Herrschaft blieb. Heute sind die Nachfahren der ersten schwedischen Immigranten die 350.000 schwedisch sprechenden Menschen in Finnland, eine Gruppe, die etwa 5 Prozent der Bevölkerung ausmacht.

Zu ihnen gehört auch meine überspannte Familie.

Mein Urgroßvater mütterlicherseits war ein ziemlich armer Bauer aus Jappo, einer kleinen Gemeinde in der Nähe der Stadt Vasa. Er hatte sechs Söhne, von denen fünf promovierten. Das sagt eine Menge über die Aufstiegschancen in Finnland aus. Ja, man bekommt Winterdepressionen und man zieht seine Schuhe aus, ehe man ein Haus betritt. Aber man kann umsonst studieren. Das ist ganz anders als hier in den USA, wo so viele Kinder mit einem Gefühl der

wachsen. Tatsächlich wurde einer dieser sechs Söhne Präsident der finnischen Zentralbank. Ein anderer war mein Großvater, Leo Waldemar Törnqvist, der mir den Computer nahe brachte.

Dann gab es da noch meinen Großvater väterlicherseits. Er war der Typ, der den Namen Torvalds ersann. Er bildete ihn aus seinem mittleren Namen. Ursprünglich hieß er Ole Torvald Elis Saxberg. Der mittlere Name für sich allein bedeutet „Thors Reich“. Mein Großvater **wurde** vaterlos geboren (Saxberg war der Mädchenname seiner Mutter) und erhielt von dem Herrn, den meine Urgroßmutter schließlich ehelichte, den Nachnamen Karanko. Farfar („Vaters Vater“) konnte den Typ so wenig leiden, dass er seinen Namen änderte. Er legte den letzten Namen ab und fügte dem Tovald ein „s“ hinzu, weil es dadurch, wie er fand, gewichtiger klang. Er hätte lieber nach etwas ganz Neuem suchen sollen, denn das hinzugefügte „s“ zerstört die Bedeutung des ursprünglichen Namens und verwirrt sowohl schwedisch als auch finnisch sprechende Menschen, die nicht wissen, wie zum Teufel sie „Torvalds“ eigentlich aussprechen sollen. Außerdem meinen sie, es müsste „Thorvalds“ heißen. Es gibt einundzwanzig Torvalds auf der Welt, und wir sind alle miteinander verwandt. Wir alle müssen mit dieser Verwirrung leben. Vielleicht bin ich deshalb im Internet immer nur „Linus“. „Torvalds“ ist einfach zu verwirrend.

Dieser Großvater unterrichtete nicht an einer Universität. Er war Journalist und Dichter. In seinem ersten Job war er Chefredakteur einer Kleinstadtzeitung in einem Ort 100 Kilometer westlich von Helsinki. Er wurde entlassen, weil er mit etwas zu großer Regelmäßigkeit im Job betrunken war. Seine Ehe mit meiner Großmutter zerbrach. Er zog in die Stadt Turku im Südwesten Finnlands, wo er wieder heiratete,

Chefredakteur der dortigen Zeitung wurde und mehrere Lyrikbände veröffentlichte, auch wenn sein Alkoholproblem ihm ein Leben lang zu schaffen machte. Wir besuchten ihn dort zu Weihnachten und zu Ostern und hatten dann auch Gelegenheit, meine Großmutter zu sehen. Farmor Märta lebt in Helsinki und ist für ihre umwerfenden Pfannkuchen berühmt.

Farfar starb vor fünf Jahren.

Okay: Ich habe nie eines seiner Bücher gelesen. Mein Vater hält es für nötig, diese Tatsache wildfremden Menschen auf die Nase zu binden.

Journalisten sind in meiner Familie Legion. Der Familiensage zufolge war einer meiner Urgroßväter, Ernst von Wendt, ein Journalist, der auf der Seite der Weißen stand und im finnischen Bürgerkrieg, der 1917 nach unserer Unabhängigkeit von Russland ausbrach, von der Roten Armee gefangen genommen wurde. (Okay, seine Bücher habe ich auch nie gelesen und man sagt mir, ich hätte nicht viel versäumt.) Mein Vater Nils (den jeder Nicke nennt) ist Fernseh- und Rundfunkjournalist und war seit seiner Studienzeit in den 1960er-Jahren in der kommunistischen Partei aktiv. Er entwickelte seine politischen Anschauungen, als er von den Gräueltaten erfuhr, die gegen kommunistische Sympathisanten in Finnland begangen wurden. Jahrzehnte später gab er zu, dass sein Enthusiasmus für den Kommunismus möglicherweise aus Naivität heraus geboren war. Er lernte meine Mutter Anna (die Mikke genannt wird) kennen, als sie beide in den 1960er-Jahren rebellische Studenten waren. Der Geschichte meines Vaters zufolge ereignete sich diese denkwürdige Begegnung bei einem Busausflug eines Clubs für schwedisch sprechende Studenten, dessen Präsident er war. Es gab da einen Rivalen, der sich ebenfalls für meine Mutter interessierte, und als sie sich für die Rückfahrt nach Helsinki fertig machten, forderte mein Vater den Rivalen auf, das Einsteigen in den Bus zu überwachen. Er nutzte die Gelegenheit, sich den Platz neben meiner Mutter zu sichern, und überredete sie, mit ihm auszugehen. (Und die Leute halten mich für das Genie in unserer Familie!)

Ich wurde mehr oder weniger zwischen Protestkundgebungen auf dem Campus geboren, wahrscheinlich während im Hintergrund Joni Mitchell lief. Das Liebesnest unserer Familie war ein Zimmer in der Wohnung meiner Großeltern. Ein Wäschekorb diente als mein erstes Kinderbettchen. Dankenswerterweise erinnert man sich für gewöhnlich nicht an diese Phase. Irgendwann um meinen 3-Monats-Geburtstag herum trat mein Vater seinen vorgeschriebenen elfmonatigen Wehrdienst an, statt als der Verweigerer aus Gewissensgründen, der er wahrscheinlich war, ins Gefängnis zu gehen. Er wurde ein so guter Soldat und ein so ausgezeichnete Schütze, dass er am Wochenende häufig Ausgang bekam. Der Familienlegende zufolge wurde meine Schwester Sara bei einem dieser ehelichen Besuche gezeugt. Wenn sie nicht von ihren zwei Blondschöpfen auf Trab gehalten wurde, arbeitete meine Mutter als Übersetzerin für die finnische Nachrichtenagentur. Jetzt arbeitet sie als Grafikerin.

Ich selbst bin dieser Mini-Dynastie des Journalismus wundersamerweise entronnen. Sara ist selbständig und übersetzt Berichte für die Nachrichten, und arbeitet darüber hinaus bei der finnischen Nachrichtenagentur. Mein Stiefbruder Leo Torvalds ist ein Videofreak und möchte als Regisseur arbeiten. Weil im Prinzip jeder in meiner Familie etwas mit Journalismus zu tun hat, kann ich es nicht lassen, Reporter damit aufzuziehen, dass ich weiß, was für ein Abschaum sie sind. Es ist mir bewusst, dass ich mich damit zum Vollidioten mache, aber im Lauf der Jahre hatten wir zu Hause in Finnland jede Menge Journalisten zu Gast, die um einer Story willen vor nichts zurückschreckten, Geschichten einfach so erfanden oder immer so wirkten, als hätten sie etwas zu viel getrunken. Okay: viel zu viel getrunken.

Bei solchen Gelegenheiten verzieht man sich besser in sein Zimmer. Wenn Mam gerade emotional schlecht drauf ist, auch. Wir wohnen in einer Drei-Zimmer-Wohnung im zweiten Stock eines nichts sagenden blassgelben Gebäudes in der Robertsgatan in Rödbergen, einem Viertel in der Nähe des Stadtzentrums von Helsinki. Sara und ihr widerlicher 16 Monate älterer Bruder teilen sich das Kinderzimmer. In der Nähe gibt es einen kleinen Park, der nach der Sinebrychoff-Familie benannt ist, die eine örtliche Brauerei besaß. Das kam mir immer komisch vor, aber eigentlich ist es auch nicht anders, als ein Basketball-Stadion nach einem Hersteller von Büroprodukten zu benennen. (Nachdem einmal eine Katze dort gesichtet wurde, hieß der Sinebrychoffpark in meiner Familie nur noch „Katzenpark“). Es gibt dort ein leer stehendes kleines Haus, in dem sich die Tauben versammeln. Der Park liegt auf einem Hügel, und im Winter kann man dort Schlitten fahren. Weitere Spielplätze sind der betonierte Hof hinter unserem Haus beziehungsweise das Flachdach des Hauses. Wenn wir Verstecken spielen, machen wir uns einen Spaß daraus, die Feuerleiter fünf Stockwerke hoch auf das Dach zu klettern.

Aber kein Spaß kann sich mit dem Spaß am Computer messen. Jetzt, wo ich den Computer zu Hause stehen habe, kann ich die ganze Nacht davor verbringen. Jeder Junge „liest“ heimlich den Playboy unter der Bettdecke. Nur ich nicht. Ich tue, als würde ich schlafen, warte, dass meine Mutter geht, springe aus dem Bett und setze mich an den Computer. Obwohl es damals noch keine Chatrooms gab.

„Linus, Zeit zum Essen!“ Manchmal kommst du nicht mal raus. Dann fängt deine Mutter an, ihren Journalistenfreunden zu erzählen,

du seist ein so pflegeleichtes Kind, dass es dich glücklich macht, wenn man dich in einer dunklen Kammer mit einem Computer abstellt und dir gelegentlich ein paar gekochte Nudeln reinwirft. Damit liegt sie nicht sehr weit daneben. Niemand befürchtete, dieses Kind könnte gekidnappt werden. (Hmmm. Ob es überhaupt jemand gemerkt hätte?) Tatsächlich waren Computer damals besser für Kinder geeignet: Sie waren weniger ausgereift als heute, und Grünschnäbel wie ich konnten unter der Haube daran herumbasteln. Heutzutage ist es mit Computern wie mit Autos: Mit zunehmender Komplexität wurde es schwieriger, sie auseinander zu nehmen und wieder zusammenzubauen und auf diese Weise zu lernen, was eigentlich dahinter steckt. Wann haben Sie zum letzten Mal etwas Schwierigeres an Ihrem Auto gemacht, als den Ölfilter zu wechseln? Statt unter den metaphorischen Hauben ihrer PCs herumzubasteln, spielen die Kinder von heute viel zu viele Computerspiele und verlieren dabei den Verstand. Nicht, dass ich etwas gegen Spiele hätte. Sie gehörten zu meinen ersten Programmen.

In einem davon steuerte man ein kleines U-Boot in einer Grotte. Es war ein Spiel wie viele andere auch. Die Welt bewegte sich seitwärts und machte ab und zu einen Schwenk, und der Spieler war das U-Boot und durfte nicht an die Wände und die Riesenfische stoßen. Das Einzige, was sich wirklich bewegte, war die Welt. Die Fische bewegten sich mit der Welt. Je länger man spielte, desto schneller bewegte sich alles. Gleichzeitig wurde die Grotte immer kleiner. Man konnte dieses Spiel nicht gewinnen. Und darum ging es mir auch nicht. Es machte Spaß, eine Woche oder so damit zu spielen, aber dann nahm ich mir das nächste Spiel vor. Das eigentlich Interessante war, den Code zu schreiben, der all das bewirkte.

Natürlich gibt es auch anderes Spielzeug, zum Beispiel Modellflugzeuge und Schiffe und Autos und Eisenbahnen. Einmal kauft Papa eine teure deutsche Modelleisenbahn. Er begründet es damit, dass er als Kind nie eine Modelleisenbahn hatte, und dass sie ein gutes VaterSohn-Hobby sein würde. Mit der Eisenbahn zu spielen, machte Spaß, aber es kam nicht im Entferntesten an die Herausforderungen heran, die ein Computer stellte. Ich habe nur ein einziges Mal Computerverbot bekommen, aber nicht, weil ich zu viel Zeit am Rechner verbrachte, sondern als Strafe für etwas anderes, wahrscheinlich Streit mit Sara. Während unserer ganzen Zeit in der High School waren wir

erbitterte Konkurrenten, vor allem, was die schulischen Leistungen anbetraf.

Der Konkurrenzkampf zahlte sich aus. Ohne meine ständigen Spötteleien wäre Sara nie motiviert gewesen, mich auszustechen und sechs Abschlussarbeiten zu schreiben anstelle der fünf, die in Finnland für den Schulabschluss erforderlich sind. Andererseits verdanke ich es Sara, dass mein Englisch nicht Grauen erregend ist. Sie hat sich immer über mein Englisch lustig gemacht, das jahrelang ein typisch finnisches Englisch war. Nebenbei bemerkt, auch meine Mutter zog mich auf -aber vor allem, weil ich wenig Interesse an den Mitschülerinnen zeigte, die Nachhilfestunden vom „Mathegenie“ wollten.

Manchmal lebten wir bei meinem Vater und seiner Freundin, manchmal lebte Sara bei Papa und ich bei Mama. Manchmal lebten wir beide bei Mama. Übrigens gibt es im Schwedischen keine Entsprechung für den Begriff „dysfunktionale Familie“. Als Folge der Scheidung hatten wir nicht besonders viel Geld. Zu meinen deutlichsten Erinnerungen an diese Zeit gehört es, dass meine Mama regelmäßig ihre einzige Geldanlage verpfänden musste - die einzige Aktie von Elisa, der Telefongesellschaft von Helsinki, die jeder bekam, der ein Telefon hatte. Die Aktie war so um die 500 Dollar wert, und wann immer es finanziell eng wurde, musste meine Mutter das Zertifikat in ein Leihhaus tragen. Ich erinnere mich daran, wie ich einmal mitkam und wie sehr ich mich dafür schämte. (Heute sitze ich im Aufsichtsrat dieser Gesellschaft. Tatsächlich ist Helsinki Telephone das einzige Unternehmen, in dem ich ein Mitglied des Aufsichtsrats bin.) Ich schämte mich auch, als Mama, nachdem ich fast das ganze Geld für meine erste Uhr gespart hatte, von mir verlangte, meinen Großvater um das restliche Geld zu bitten.

Es gab eine Zeit, in der meine Mutter abends arbeitete, so dass Sara und ich uns selbst etwas zum Abendessen beschaffen mussten. **Eigentlich sollten** wir in den Laden an der Ecke gehen und uns etwas zu Essen kaufen. Stattdessen holten wir uns Süßigkeiten, und es war toll, bis spätabends aufzubleiben und am Computer zu sitzen. Andere Jungen hätten unter solchen Umständen den Playboy auf der Bettdecke „gelesen“.

Kurz nachdem mein Großvater starb, fühlte

m der Lage, weiter selbst für sich zu sorgen. Sie lag zehn Jahre im Krankenhaus mit etwas, was sie „Benommenheit“ nannte. Nach ein

paar Jahren zogen wir in ihre Wohnung ein. Sie war im ersten Stock eines soliden Altbaus gelegen, der noch aus der russischen Ära stammte, in der Petersgatan, in der Nähe des wundervollen Parks an der Uferpromenade von Helsinki. Die Wohnung hatte eine kleine Küche und drei Schlafzimmer. Sara bekam das größte Zimmer. Der hochaufgeschossene Teenager, der mit einer dunklen Kammer und einer gelegentlichen Portion Nudeln glücklich war, zog in das kleinste ein. Ich hängte dicke schwarze Vorhänge auf, so dass kein Lichtstrahl hereindringen konnte. Der Computer fand seine Heimstatt auf einem winzigen Schreibtisch vor dem Fenster, etwa sechzig Zentimeter von meinem Bett entfernt.

Der Name Linus Torvalds war mir vage ein Begriff, als mich ein Redakteur des Sonntagsmagazin der San Jose Mercury News im Frühjahr 1999 aufforderte, ein Porträt über ihn zu schreiben. Linux war ein Jahr davor zum Schlagwort geworden, als eine Reihe von Unternehmen, allen voran Netscape, das Konzept des Open-Source-Codes beziehungsweise das Betriebssystem selbst übernommen hatten. Nicht, dass ich über diese Entwicklungen auf dem Laufenden gewesen wäre. Anfang der 1990er Jahre war ich Herausgeber eines Magazins gewesen, das sich mit Unix und Open-Source-Fragen beschäftigte. Ich konnte mich deshalb dunkel daran erinnern, irgendwo gelesen zu haben, Linus sei ein finnischer Student, der in seiner Studentenbude eine mächtige Unix-Version geschrieben hatte und sie kostenlos über das Internet verteilte. Das entsprach nicht mehr ganz den Tatsachen. Der Redakteur rief mich an, weil Linus gerade die gefeierte - und atackierte - Hauptattraktion bei einer Linux-Messe in San Jose gewesen war. Diese Tatsache veranlasste den Herausgeber, mich mit den dorten für das Projekt zu gewinnen: „Wir haben direkt hier in, äh, Santa Clara einen globalen Superstar sitzen.“ Er faxte mir ein paar Zeitungsberichte zu.

Linus lebte damals seit zwei Jahren im Silicon Valley und arbeitete für die Transmeta Corporation, um die damals ein ziemliches Geheimnis gemacht wurde. Transmeta entwickelte seit Jahren einen Mikroprozessor, der versprach, die Computerindustrie auf den Kopf zu stellen. Linus hatte dort einen Job, der es ihm erlaubte, seiner zeitraubenden Aufgabe als ultimatives Oberhaupt von Linux und letzte Autorität hinsichtlich aller Änderungen des Betriebssystems nachzukommen. (Tatsächlich hatten seine Anhänger die rechtlichen Schritte eingeleitet, ihn zum gesetzlichen Eigentümer des Warenzeichens Linux zu machen.) Und er hatte Zeit, als Aushängeschild der aufblühenden Open-Source-Bewegung über den Globus zu reisen. Irgendwie war er eine Art rätselhafter Volksheld geworden. Während Bill Gates, jedermanns bevorzugte Zielscheibe, in Pomp und Prunk in

seinem Xanadu am Lake Washington lebte, wohnte Linus mit seiner Frau und seinen kleinen Töchtern in einer engen Doppelhaushälfte in Santa Clara. Anscheinend lief ihn der Geldregen, der auf Scharen weniger talentierter Programmierer herniederging, völlig unbeeindruckt. Und allein seine Anwesenheit warf unter den Aktien- und Optionen-hörigen Zuarbeitern des Silicon Valley ein unlösbares Rätsel auf, nämlich: Wie konnte ein so brillanter Kopf nur so wenig Interesse daran haben, reich zu werden?

Linus hat keinen Mitarbeiterstab, hört den Anrufbeantworter nicht ab und reagiert selten auf E-Mails. Es dauerte Wochen, ihn ans Telefon zu bekommen, aber als es mir endlich gelang, stimmte er bereitwillig einem Interview zu dem für ihn frühestmöglichen Zeitpunkt zu, nämlich etwa einen Monat später: im Mai 1999. Da ich meine Interviewpartner leidenschaftlich gern in kompromittierenden Situationen befrage, kam ich auf die Idee, eine finnische Sauna könnte den perfekten Hintergrund für das Porträt bilden. In einem gemieteten MustangCabrio, mit einem Photographen am Steuer, fuhren wir hinüber nach Santa Cruz, wo sich auf dem Areal eines New-AgeNudistenclubs die angeblich beste Sauna der Bay Area befand.

Er war mit einer geöffneten Cola-Dose bewaffnet, als er aus den Tiefen der Transmeta-Büros in einem gesichtslosen Büropark in Santa Clara auftauchte. Er trug die typische Programmiereruniform aus Jeans, Tagungs-T-Shirt und der unvermeidlichen Socken-und-Sandalen-Kombination, die er angeblich bereits bevorzugt hatte, bevor er jemals einem anderen Programmierer begegnet war. „Das muss ein Programmierernaturgesetz sein“, behauptete er, als ich seine Fussbekleidung ins Gespräch brachte.

Meine erste Frage an Linus, als wir auf dem Rücksitz Platz genommen hatten, war ein Aufwärmer. „Kommen Sie aus einer Technikerfamilie?“ fragte ich und fingerte an meinem Kassettenrekorder herum.

„Nein, sie haben alle etwas mit Journalismus zu tun“, antwortete er und fügte hinzu: „Deshalb weiß ich, was für ein Abschaum ihr seid.“

Er konnte nicht im Ernst glauben, dass er damit durchkommen würde.

„Oh. Sie stammen von Abschaum ab?“ gab ich zurück.

Der beste Programmierer der Welt lachte so sehr, dass er eine ColaFontäne auf den Nacken des Photographen und Fahrers prustete. Er lief rot an. Das war der Beginn eines denkwürdigen Nachmittags.

Aber es kam noch bizarrer. Finnen sind fanatische Saunagänger, und das war sein erster Saunabesuch seit fast drei Jahren. Der blasse nackte Superstar saß mit angelaufener Brille auf der höchsten Bank; **sein nasses gelbbraunes Haar** klebte ihm im Gesicht und der **Schweif rann in Sturzbächen** den Körperteil hinunter, den ich später aus reiner Gutmütigkeit als seinen „beginnenden Bauchansatz“ bezeichnen sollte. Er war von gebräunten, körperbesessenen Santa Cruzianern und ihrem monotonen New-Age-Gerede umgeben, aber er schien über **alledem** zu stehen und erklärte mir hingerissen die authentische Bauweise der Sauna. Ein seliges Lächeln breitete sich auf seinem Gesicht aus. Ich bin davon überzeugt, dass die meisten Leute im Silicon Valley glücklicher sein müssen als irgendjemand sonst auf der Welt. Zum einen sitzen sie am Steuerpult der wirtschaftlichen Revolution. Vor, **allem** aber werden sie alle unerträglich reich, ganz gleich, ob sie dem neuen oder dem alten Valley angehören. Aber nie sieht man sie **lächeln**, jedenfalls nicht außerhalb der Büros ihrer Broker.

Die meisten anerkannten Technologiespezialisten - und die meisten **nicht** anerkannten auch - sind von dem dringenden Wunsch beseelt, dich wissen zu lassen, wie brillant sie sind. Und dass sie die **Protagonisten einer** Mission sind, die weitaus wichtiger ist als, sagen wir, der Kampf um den Weltfrieden. Bei Linus war das anders. Dass ihm jede Form von Selbstbeweihräucherung fremd war, wirkte ausgesprochen entwaffnend und lief ihn inmitten der bombastischen Elite des Silicon Valley ungemein sympathisch erscheinen. Linus schien über allen zu stehen. Über den New Agern. Über den Hightech-Milliardären. Er **kam mir** nicht wie ein Rentier vor, das im Bannstrahl des globalen Scheinwerferlichts steht, sondern eher wie ein zauberhafter Alien, der auf die Erde gebeamt wurde, um uns die Verrücktheit unserer selbstsüchtigen Lebensweise vor Augen zu führen.

Und ich gewann den Eindruck, dass es sich nicht besonders für ihn rechnete. Linus hatte erwähnt, dass ein wichtiger Teil des Saunarituals darin bestand, hinterher zusammen herumsitzten, Bier zu trinken und über Gott und die Welt zu diskutieren. Als Vorbereitung darauf hatten wir Dosen mit Fosters-Bier in ein paar Büschen versteckt. Wir schnappten uns das Bier und liefen uns in dem „ruhigen“ Hot-Tub nieder, wo wir die Fosters-Dosen öffneten, während der Photograph seine Bilder

V.

Im Prinzip tat ich vier Jahre lang nichts anderes, als vor einem Computer zu sitzen.

Okay, da war die Schule: Norssen High, die von den fünf Helsinkier High-Schools für schwedisch sprechende Schüler am zentralsten und unserer Wohnung am nächsten gelegen war. Mathe und Physik waren interessant und daher einfach. Aber wann immer ein Fach reines Auswendiglernen erforderte, sank mein Enthusiasmus in den Keller. Geschichte war langweilig, wenn es darum ging, das Datum der Schlacht von Hastings zu wissen, und interessant, wenn wir die Wirtschaftsfaktoren eines Landes durchnahmen. In Geografie war es das Gleiche. Ich meine, wen *kümmert* es schon, wie viele Menschen in Bangladesch leben? Gut, bei näherer Betrachtung ist es vielleicht für eine ganze Menge Leute wichtig. Aber ich driftete jedenfalls weniger leicht in Tagträume über meinen Computer ab, wenn wir etwas Fesselnderes als Statistiken durchnahmen und zum Beispiel die Monsunwinde oder die Gründe dafür besprachen.

Der Sportunterricht stand auf einem anderen Blatt. Ich erzähle Ihnen vermutlich nichts Neues, wenn ich enthülle, dass ich nicht eben der sportlichste Junge auf der Fennoskandia-Halbinsel war. Außerdem war ich damals, ob Sie es glauben oder nicht, dünn. Turnen ging gerade noch. Aber wenn wir Fußball oder Eishockey spielten, war es für mich an der Zeit, blau zu machen.

Das zeigte sich an meinen Noten. In Finnland gibt es Noten zwischen vier und zehn. Also hatte ich Zehner und ein paar Neuner in Mathe, Physik, Biologie und dem ganzen Rest. Aber eine Sieben in Sport. Einmal sogar eine Sechs. Und ich hatte eine Sechs in Werken. Das gehörte auch nicht zu meinen Stärken. Anderen Jungs sind schön gearbeitete

Serviettenhalter oder Hocker als Erinnerung an den Werkunterricht geblieben. Mir sind nichts als ein paar Holzsplitter geblieben, die noch heute in meinem Daumen stecken. Vielleicht sollte ich an dieser Stelle erwähnen, dass die wunderschöne Schaukel in unserem Garten, auf der meine Töchter so viele glückliche Stunden verbringen, mein Schwiegervater gebaut hat.

Meine High-School war keine dieser Anstalten für außergewöhnlich begabte oder ehrgeizige Kinder, die Art von Schule, wie es sie in den meisten amerikanischen Städten gibt. Solche Schulen laufen dem finnischen Selbstverständnis zuwider. Man sondert die guten Schüler nicht aus - oder die schlechten, wenn man so will. Allerdings hatte jede Schule ihr Spezialgebiet, ein Fach, das nicht Pflicht war, aber an keiner anderen Schule gelehrt wurde. An der Norssen High School war das Latein. Und Latein machte mir Spaß. Mehr Spaß als Finnisch oder Englisch.

Zu dumm, dass es eine tote Sprache ist. Ich würde mich liebend gern mit ein paar Kumpels treffen und in Latein Witze erzählen oder Entwurfsstrategien für Betriebssysteme diskutieren.

Es machte auch Spaß, in dem Cafe in der Nähe der Schule herumzuhängen. Es war ein Treff für bestimmte Leute, vor allem die, die nicht heimlich hinter der Schule rauchten. Dort gingen wir hin, wenn wir den Sportunterricht schwänzten oder wenn wir eine Zwischenstunde hatten, was manchmal vorkam.

Das Cafe war seit den Tagen des Rechenschiebers ein Zufluchtsort für Geeks. Es war auch der einzige Ort, wo Schüler etwas anschreiben lassen konnten. Das hieß, man konnte sich etwas bestellen, und es wurde auf einer Liste notiert, auf der alles stand, was man gegessen oder getrunken hatte, und wenn man mal zufällig Geld hatte, beglich man die Rechnung. Wie ich den finnischen Technologiewahn kenne, verwenden sie inzwischen eine Datenbank dafür.

Meine Bestellung war immer die gleiche: Eine Coke und ein Doughnut.

So jung und schon so ernährungsbewusst.

Im Großen und Ganzen war ich besser in der Schule als meine Schwester Sara, die kontaktfreudiger, hübscher anzusehen und netter zu den Leuten war und die, das sollte ich hinzufügen, den Auftrag hat, dieses Buch ins Schwedische zu übersetzen. Doch am Ende hat sie mich übertroffen, weil sie Prüfungen in mehr Fächern ablegte.

Tatsächlich brachte ich Mädchen nur nach Hause mit, wenn sie unbedingt Nachhilfestunden von mir wollten. Das kam nicht besonders oft vor und war nie meine Idee, aber mein Vater glaubt bis heute, dass sie nicht nur an Mathenachhilfe interessiert waren. (Er bildet sich ein, sie wären irgendwie auf seine Gleichung *Stattliche blase* = *Stattlicher Mann* hereingefallen.) Wenn sie auf diese Art von Action aus gewesen sein sollten, hatten sie keinen willigen Partner. Ich meine, ich kam noch nicht einmal darauf, was sie mit „Heavy Petting“ meinten. Ich hatte öfter die 15 Pfund schwere Katze eines Nachbarn versorgt und konnte mir nicht vorstellen, was so toll daran sein sollte, eine schwergewichtiges Haustier zu streicheln.

Ja, ich war definitiv ein Geek. Keine Frage. Und das in einer Zeit, bevor Geeks als sexy galten. Oder jedenfalls als hip. Dazu kam, dass ich nicht nur ein Geek, sondern außerdem schüchtern war - oder ist das redundant?

Und so saß ich vor meinem Computer und war wunschlos glücklich. In Finnland trägt man bei der Abschlussfeier der High-School einen weißen weichen Hut mit schwarzem Band. Es gibt eine Zeremonie, in der sie dir dein Zeugnis überreichen, und wenn du nach Hause kommst, sind alle Verwandten da, mit Unmengen von Champagner, Blumen und Kuchen. Außerdem gibt es eine Party für die ganze Klasse in einem Restaurant. So war es auch bei mir, und ich nehme an, es hat mir Spaß gemacht, aber ich kann mich an die Einzelheiten nicht mehr erinnern. Aber fragen Sie mich nach den technischen Daten meines 68008-Prozessors, und ich kann sie Ihnen aufsagen, als hätte ich das absolute Gedächtnis.

VI.

Mein erstes Jahr an der Uni verlief erstaunlich produktiv. Ich schaffte es, die Scheine - die in Finnland „Studien-Wochen“ genannt werden - zu bekommen, die man bekommen sollte. Es war das einzige Jahr, in dem mir das gelang. Vielleicht war es die Begeisterung über die neue Umgebung oder die Gelegenheit, tief in die Themen einzutauchen, oder vielleicht war es einfach bequemer zu studieren, als am Studentenleben teilzunehmen und sich mit ritueller Regelmäßigkeit über seine Freunde auszukotzen. Ich weiß nicht, was an meinen zufriedenstellenden Leistungen in diesem ersten Jahr schuld war. Aber seien Sie versichert, es kam nicht wieder vor. Meine akademische Karrierekurve verlief steil nach unten.

Zu diesem Zeitpunkt stand mein Hauptfach noch nicht fest. Später belegte ich im Hauptfach Informatik und im Nebenfach Physik und Mathematik. Eines der Probleme war, dass an der gesamten Universität Helsinki nur noch ein weiterer schwedisch sprechender Student, Lars Wirzenius, Informatik als Hauptfach studieren wollte. Wir traten beide dem Spektrum bei, dem Club für schwedisch sprechende Studenten, die ein naturwissenschaftliches Fach studierten, und dort hatten wir, wie sich herausstellen sollte, eine Menge Spaß. Der Club bestand aus Studenten der harten Naturwissenschaften wie Physik oder Chemie. Im Klartext: Es gab dort nur Jungs.

Aber immerhin teilten wir uns den Raum mit der entsprechenden Organisation für schwedisch sprechende Studenten der weicheren Wissenschaften wie Biologie und Psychologie. Auf diese Weise trafen wir auf weibliche Wesen, so unwohl sich manche von uns dabei auch **fühlen mochten**. Okay, alle von uns.

Spektrum ähnelte in vieler Hinsicht einer Studentenverbindung im amerikanischen Stil, aber du musstest nicht mit den anderen Typen

zusammenleben oder dich mit jemandem abgeben, der sich nicht für Naturwissenschaften interessierte. Wir hatten regelmäßige Treffen am Mittwohabend, wo ich lernte, was der Unterschied zwischen einem Pils und einem Hellen war. Bei seltenen Gelegenheiten gab es einen Wettbewerb, wer am meisten Wodka vertragen konnte. Aber vieles davon passierte erst in den späteren Jahren meiner Unikarriere. Zeit dafür war reichlich vorhanden: Ich studierte acht Jahre und hatte am Ende nichts als einen Master-Abschluss vorzuweisen (die Ehrendoktorwürde, die mir die Universität im Juni 2000 verlieh, nicht mitgerechnet.)

Wenn ich an dieses erste Jahr zurückdenke, erinnere ich mich vage an Straßenbahnfahrten zwischen der Uni und meinem Zimmer, wo sich Bücher und Computerzubehör stapelten. Ich lag im Bett und las einen Science-Fiction-Roman von Douglas Adams, warf ihn auf den Boden und griff nach einem Physikbuch, wälzte mich aus dem Bett und schrieb am Computer ein Programm für ein neues Spiel. Die Küche war gleich neben meinem Zimmer und gelegentlich holte ich mir dort Kaffee oder Cornflakes.

Vielleicht ist deine Schwester irgendwo in der Nähe oder vielleicht ist sie mit Freunden weggegangen. Oder vielleicht wohnt sie zurzeit gerade mal bei deinem Vater. Vielleicht ist deine Mutter da oder vielleicht ist sie im Büro oder vielleicht ist sie mit ihren Journalistenfreunden weggegangen. Oder ein Freund ist vorbeigekommen, und ihr habt euch in der Küche eingenistet, wo ihr literweise Tee trinkt, auf MTV *Bevis and Butthead* in Englisch anschaut und überlegt, ob ihr irgendwo hingehen solltet, um Billard zu spielen, aber draußen ist es zu kalt.

Und glücklicherweise gibt es in diesem Lebensabschnitt keinen Sportunterricht.

Den wird es erst im nächsten Jahr wieder geben. Das ganze Jahr über. Wenn die finnische Armee jeden jungen Mann einzieht. Viele Jungs leisten ihren Wehrdienst gleich nach der High-School ab. Mir war es sinnvoller erschienen, damit zu warten, bis ich ein Jahr an der Uni hinter mir hatte.

In Finnland hat man die Wahl: Entweder du gehst acht Monate zur Armee oder ein Jahr lang zum Zivildienst. Wenn du starke religiöse Gründe oder eine andere stichhaltige Entschuldigung geltend machen kannst, bleibt dir beides erspart. Für mich gab es diesen Ausweg

nicht. Und der Zivildienst schien mir auch keine gute Alternative zu sein.

Das lag nicht daran, dass ich etwas dagegen gehabt hätte, Menschen zu helfen. Es lag wahrscheinlich eher an der Angst, dass sich der Dienst am Nächsten womöglich als *noch* langweiliger erweisen würde als der Dienst mit der Waffe. Ich kann nicht glauben, dass ich das so offen sage. Aber jeder, der sich für den Zivildienst entschieden hat, wird Ihnen sagen, dass man sich entweder im Voraus um eine gute Aufgabe kümmern muss oder irgendeine uninteressante Aufgabe zugeschoben bekommt. Ich konnte auch nicht aus Gewissensgründen verweigern. Auch wenn ich nicht das Geringste dagegen gehabt hätte, meine vaterländische Pflicht zu verweigern, ist und bleibt es eine Tatsache, dass ich tatsächlich ein Gewissen habe: Wenn es hart auf hart käme, würden mich keine unüberwindlichen Überzeugungen davon abhalten, ein Gewehr anzufassen oder Menschen zu töten.

Wenn du dich also für die Armee entscheidest, stehen dir wieder zwei Möglichkeiten offen. Entweder du reißt als 08/15-Gefreiter die obligatorischen acht Monate herunter oder du wirst Offizier und leistest elf Monate ab. Mir schwante, die Offiziersausbildung könnte möglicherweise etwas interessanter sein - trotz der zusätzlichen 129600 Minuten. Sie würde sich auch besser rechnen.

Und so kam es, dass Ihr (damals) 60 Kilogramm schwerer Held in der finnischen Armee Leutnant der Reserve wurde. Mein Job war die Feuerleitung. Dafür brauchst du kein Genie zu sein. Du bekommst die Koordinaten für die schweren Geschütze mitgeteilt. Du bestimmst mit **Hilfe der Landkarte**, wo ihr seid, und triangulierst, wohin ihr schießen **sollt**. Du berechnest die Koordinaten und funkst sie oder kommunizierst sie über die Telefonleitung, die du vorher mit den anderen verlegt hast. Du sagst den Schützen, wohin sie zielen sollen.

Ich erinnere mich daran, vor Antritt des Wehrdienstes sehr nervös gewesen zu sein, weil ich nicht wusste, was mich erwartete. Manche Leute hatten ältere Brüder oder jemanden, mit dem sie über die Armee reden konnten und wussten deshalb, womit sie rechnen mussten. Mir sagte niemand, was Sache war. Gut, jedem ist irgendwie klar, dass der Wehrdienst kein Spaß ist. Das erzählt jeder, der einmal dort war. Aber **ich hatte keine** klare Vorstellung, wie es wirklich sein würde, und das machte mich nervös. Wenn ich Leuten dieses Buch zum Lesen gebe, **fühle** ich mich ähnlich.

Zu den härtesten Zeiten in der Armee gehörte es, in den Wäldern von Lappland mit Kabeln herumzulaufen, die tonnenschwer an uns zu hängen schienen. Ehrlich gesagt, ich glaube, sie *waren* tonnenschwer. Während der Grundausbildung, die auch Offiziersanwärter ableisten müssen, befiehlt man dir, dir eine riesige Kabelrolle auf den Bauch und zwei weitere auf den Rücken zu packen und damit zwanzig wahnwitzige Kilometer zu marschieren. Dann wieder stehst du einfach nur rum und wartest darauf, dass etwas passiert.

Oder du läufst stundenlang auf Skiern durchs Gelände, bis du endlich den Platz erreichst, wo das Zelt aufgebaut werden soll. Damals wurde mir klar: Wenn Gott gewollt hätte, dass wir Ski fahren, hätte Er/Sie/Es uns mit länglichen Fiberglaskufen statt mit Füßen ausgestattet. Warten Sie eine Sekunde, ich glaube nicht unbedingt an Gott.

Dann musst du das Zelt aufbauen und das Feuer schüren, bevor du essen kannst. Dir ist kalt und du bist hungrig und müde, weil du seit zwei Tagen nicht geschlafen hast. Ich weiß, dass es Leute gibt, die gutes Geld dafür zahlen, um an solch extremen Outdoor-Abenteuern als „charakterstärkende Erfahrung“ teilzunehmen. Sie sollten einfach zur finnischen Armee gehen.

Eigentlich gab es diese Outdoor-Marathons nicht oft. Aber es gab sie. Nach meiner Rechnung verbrachten wir von den elf Monaten über 100 Tage in den Wäldern. Und Wälder gibt es in Finnland reichlich: 70 Prozent des Landes sind mit Wald bedeckt. Ich hatte das Gefühl, jeden Einzelnen persönlich kennen zu lernen.

Als Offizier war ich für die Feuerleitung in einer Fünfergruppe verantwortlich. Das bedeutet einfach, dass du wissen sollst, wo es langgeht, und die Sache komplizierter erscheinen lässt, als sie eigentlich ist. Aber alles in allem war es ziemlich uninteressant, und ich war kein sehr guter Anführer. Und es lag mir nicht, Befehle zu erteilen. Ich konnte sie einigermaßen entgegennehmen - der Trick dabei ist, dass du es nicht persönlich nimmst -, aber ich sah meine Lebensaufgabe nicht darin, der Beste zu sein.

Jedenfalls damals nicht.

Habe ich erwähnt, wie kalt es in Lappland werden kann?

Wenn ich so darüber nachdenke, habe ich den Wehrdienst gehasst. Aber wie das Leben so spielt: Als er vorbei war, erschien er mir schlagartig als eine wunderbare Erfahrung.

Außerdem habe ich seither ein Thema, das ich bis ans Ende meiner Tage mit praktisch jedem Finnen erörtern kann. Tatsächlich behaupten manche Leute, die Wehrpflicht sei hauptsächlich dazu da, dass die finnischen Männer für den Rest ihres Lebens etwas haben, worüber sie beim Bier reden können. Alle haben die gleiche elende Erfahrung hinter sich gebracht. Sie hassten die Zeit bei der Armee, um hinterher mit großem Vergnügen darüber zu reden.

VII.

Lassen Sie mich, wenn wir schon gerade beim Thema sind, noch ein bisschen mehr von Finnland erzählen. Bei uns gibt es wahrscheinlich mehr Rentiere als irgendwo sonst auf der Welt. Und auch Alkoholiker und Tangotänzer sind bei uns reichlich vertreten. Verbringen Sie einen Winter in Finnland, und Sie wissen, warum es so viele Trinker gibt. Für die Tangoistas gibt es keine Entschuldigung, aber glücklicherweise findet man sie hauptsächlich in Kleinstädten, wo man ihnen nie zu begegnen braucht.

Einer kürzlich durchgeführten Studie zufolge sind die finnischen Männer die potentesten in Europa. Es muss all das Rentierfleisch sein, oder vielleicht liegt es auch an den Stunden, die wir in der Sauna verbringen. In diesem Land gibt es buchstäblich mehr Saunas als Autos. Niemand weiß eigentlich, wie diese Religion entstanden ist, aber es ist Tradition, zumindest in manchen Orten, erst die Sauna und dann das Haus zu bauen. Viele Mehrfamilienhäuser haben eine Sauna im ersten Stock oder im Dachgeschoss, und jede Familie bekommt ihre eigene private Saunastunde zugeteilt - zum Beispiel donnerstags von 19 bis 20 Uhr. (Donnerstage und Freitage sind die typischen Saunatage.) Auf diese Weise müssen Sie nicht den Horror ertragen, Ihre Nachbarn nackt zu sehen. Ich habe einmal einen englischsprachigen Reiseführer über Finnland durchgeblättert, der den Leser ausführlich darüber belehrte, die Finnen hätten niemals Sex in der Sauna und wären schockiert, wenn sie erführen, eine solche Entweihung habe stattgefunden und sei es auch nur in der Phantasie des Touristen. Ich konnte mich vor Lachen kaum halten, als ich das las, denn die Sauna ist in einem finnischen Haushalt ein so neutraler Ort, dass das Buch genauso gut davon hätte abraten können, auf dem Küchenfußboden Sex zu machen. Ich finde, das Thema wird allzu sehr hochgespielt. In

abgelegenen Orten werden sogar Babys in der Sauna zur Welt gebracht, weil es nur dort warmes Wasser gibt, und es gibt hier und da den Brauch, zum Sterben in die Sauna zu gehen. Diese Regeln gelten übrigens nicht in meiner Familie, wo man der ganzen Sache zurückhaltend gegenüber steht.

Es gibt noch andere Eigenschaften, die Finnen von anderen Vertretern der menschlichen Spezies unterscheiden. Eine davon ist dieses traditionelle Schweigen. Niemand redet viel. Man steht einfach so herum und sagt nichts. Auch diese Regel gilt nicht für meine Familie, die ich großzügig als „exzentrisch“ beschreiben möchte.

Finnen sind stoisch bis zum Umfallen. Möglicherweise verdanken wir es unserem stillen Dulden und unserer wilden Entschlossenheit, dass wir die Unterdrückung durch Russland, eine Folge blutiger Kriege und ein Wetter, das zum Himmel stinkt, überstanden haben. Aber heutzutage

wirkt so viel Gleichmut einfach nur seltsam. Der Schriftsteller Bertolt Brecht verbrachte die Jahre des Zweiten Weltkriegs in Helsinki, und von ihm stammt der berühmte Spruch, die Besucher eines Bahnhofscafes hätten „in zwei Sprachen geschwiegen“. Er nutzte die erstbeste Gelegenheit, über Wladiwostok in die USA auszureisen.

Selbst heute finden Sie, wenn Sie in einer finnischen Stadt - vor allem einer kleineren - eine Bar betreten, sehr wahrscheinlich reglose Männer vor, die allein an einem Tisch sitzen und in die Luft starren. In Finnland respektiert man die Privatsphäre der anderen - auch das ist eine wichtige Sache bei uns - und niemand käme auf die Idee, ein Gespräch anzufangen. Es ist ein Rätsel, denn eigentlich sind die Finnen ziemlich freundliche Menschen. Aber wenigen Menschen ist es vergönnt, das herauszufinden.

Soviel ich weiß, geht es in den finnischen Lesbenbars wesentlich ungezwungener zu.

Da wir Finnen es hassen, von Angesicht zu Angesicht miteinander zu reden, bilden wir den idealen Markt für Mobiltelefone. Wir haben die neuen Geräte mit einem Enthusiasmus übernommen, der von keiner anderen Nation übertroffen wurde. Es ist ungeklärt, welches Land sich der meisten Rentiere pro Kopf rühmen kann - bei näherer Betrachtung könnte dieser Rekord auch an Norwegen gehen -, aber es steht außer Frage, welche Nation auf der Welt die meisten Handys pro Mann, Frau und Kind vorzuweisen hat. In Finnland wird gerade überlegt, sie bei der Geburt gleich fest am Körper anzubringen.

Und wir wissen eine Menge damit anzufangen. Die Finnen schicken sich gewohnheitsmäßig Textmeldungen oder nutzen Handys als Möglichkeit, bei Klassenarbeiten zu schummeln (schicke einem Freund eine Frage und warte darauf, dass er dir ein SMS zurücksendet). Wir setzen die Taschenrechnerfunktion ein, von der die meisten Amerikaner noch nicht einmal wissen, dass sie auf einem Handy vorhanden ist. Als Nächstes werden vermutlich Cafebesucher die Nummer der einsamen Person am Nebentisch wählen und per Handy ein Gespräch anfangen. Selbst wenn man den phänomenalen Erfolg von Nokia außer Acht lässt, haben Mobiltelefone Finnland verändert wie nichts anderes seit der längst vergessenen Einführung der legendären Sauna.

Eigentlich ist es keine Überraschung, dass Mobiltelefone in Finnland so bereitwillig angenommen wurden. Das Land machte sich neue Technologien von jeher schnell und vertrauensvoll zu Eigen. Beispielsweise erledigen die Finnen anders als jede andere Nation auf der Welt Überweisungen und alle Bankgeschäfte routinemäßig elektronisch -kein Vergleich zu diesem mickrigen Pseudo-Electronic-Banking, wie es in den USA üblich ist. In Finnland gibt es mehr Internetanschlüsse pro Kopf als in jedem anderen Land. Manche führen diese Aufgeschlossenheit gegenüber technischen Neuerungen auf das starke Bildungssystem zurück - in Finnland gibt es weniger Analphabeten als irgendwo sonst auf der Welt, und es gibt keine Studiengebühren, so dass der typische Studierende sechs oder sieben Jahre an der Uni bleibt. Oder acht Jahre, so wie ich. Man kommt nicht umhin, etwas zu lernen, wenn man einen so großen Teil seines Lebens an der Uni herumhängt. Andere sagen, der technologische Vorsprung gehe auf die Zeit der Infrastrukturverbesserungen im Transportwesen zurück, die aufgrund der an Russland zu zahlenden Kriegsreparationen erforderlich wurden. Wieder andere behaupten, er habe etwas mit der (manchmal unerträglich) homogenen Bevölkerung zu tun.

Linus und ich sitzen am Esstisch. Wir sind gerade von einem Freizeitpark zurückgekehrt, wo man Autorennen fahren und Baseball-Bälle ins Netz schlagen kann. Tove räumt Lebensmittel weg, Patricia und Daniela streiten sich um ein Buch, das ich einer von ihnen mitgebracht habe. Ich schiebe einen Plüschpinguin und ein riesengroßes Glas Erdnussbutter beiseite, schalte den Kassettenrekorder ein und bitte Linus, etwas über seine Kindheit zu erzählen.

„Eigentlich erinnere ich mich nicht sehr gut an meine Kindheit“, sagt er abwehrend.

„Wie ist das möglich? Sie liegt doch erst ein paar Jahre zurück?“

„Frag Tove. Ich hab ein lausiges Gedächtnis für Namen oder Gesichter oder was ich getan habe. Ich muss sie nach unseren Telefonnummern fragen. Ich erinnere mich an Regeln oder wie Dinge organisiert sind, aber ich kann mich nie an Einzelheiten erinnern, und ich erinnere mich nicht an die Einzelheiten meiner Kindheit. Ich erinnere mich nicht daran, wie etwas passiert ist oder was ich gedacht habe, als ich klein war.“

„Hmm, hattest du zum Beispiel Freunde?“

„Ein paar. Ich war nie sehr kontaktfreudig. Ich bin heute viel, viel kontaktfreudiger als damals.“

„Und wie war es so? Ich meine, Erinnerst du dich daran, dass du an einem Sonntagmorgen aufgewacht bist und mit deiner Schwester und deinen Eltern irgendwo hingefahren bist?“

„Meine Eltern lebten damals getrennt.“

„Wie alt warst du, als sie sich trennten?“

„Weiß nicht. Sechs vielleicht. Oder zehn. Ich weiß es nicht mehr.“

„Was war mit Weihnachten? Erinnerst du dich an Weihnachten?“

„Hm, ich erinnere mich dunkel daran, dass wir uns schön anzogen und meinen Großvater in Turku besuchten. An Ostern auch. Sonst erinnere ich mich eigentlich an nichts.“

„Und wie war das mit deinem ersten Computer?“

„Das war der berühmte VIC-20, den mein Großvater mütterlicherseits gekauft hat. Er war in eine Schachtel eingepackt.“

„Wie groß war die Schachtel? Ungefähr so groß, dass ein Paar Skistiefel reingepasst hätten?“

„ja, so ungefähr.“

„Und was war mit deinem Großvater? Kannst du dich noch gut an ihn erinnern?“

„Er war wahrscheinlich mein nächster Verwandter, aber ich kann mich nicht ... Okay. Er hatte Übergewicht, aber er war nicht dick. Er bekam eine Glatze. Er war in sich gekehrt, so eine Art geistesabwesender Professor, der er ja auch war. Ich saß immer auf seinem Schoß und tippte seine Programme ein.“

„Kannst du dich daran erinnern, wonach er gerochen hat?“

„Nein. Was soll denn diese Fragerlei?“

„Alle Großväter riechen nach irgendetwas. Billiges Rasierwasser. Cognac. Zigarren. Wonach hat er gerochen?“

„Wei/3 ich nicht. Ist mir nicht aufgefallen. Ich war zu sehr mit dem Computer beschäftigt.“

Geburt eines Betriebssystems



*Achtung: Mittelschwere Geek-Sprache bis Seite 128.

Manche Leute sortieren ihre Erinnerungen nach den Autos, die sie fuhren, den Jobs, die sie hatten, den Orten, wo sie lebten, oder den Mädchen, mit denen sie zusammen waren. Ich mache meine Lebensphasen an Computern fest. Ich hatte in meiner Schul- und Studentenzeit nur drei Computer. Da gab es, wie gesagt, den von meinem Großvater geerbten VIC-20. Er war einer **der ersten Heimcomputer und gehörte damit zu den Vorläufern der heutigen PCs**. Der Commodore 64 wurde eine Art großer Bruder des VIC-20, gefolgt vom Amiga, der vor allem in Europa eine große Anhängerschaft hatte. Aber anders als der PC oder sogar der **Apple II, der sich bereits durchgesetzt hatte, als ich mit dem VIC herumspielte, wurden diese Computer nie wirklich populär**. **Damals, vor der Verbreitung der PCs, wurden die meisten Programme auf Heimcomputern in Assemblersprache geschrieben. (Ich kann nicht glauben, dass ich mir angewöhnt habe, meine Sätze mit „Damals ...“ zu beginnen.) Computer hatten ihr eigenes selbst gestricktes Betriebssystem, das Gegenstück dessen, was DOS für den PC war.** Je nach Computer war es mehr oder weniger anspruchsvoll gestrickt. **Wie DOS hatte das Betriebssystem einen Programmlader und eine Sprachumgebung für BASIC.** Damals gab es noch keine Standards, und eine Reihe von Firmen wollten den Markt beherrschen. Der Amiga war eines der bekannteren Modelle. **Als ich aus dem VIC-20 herausgeholt hatte, was ich konnte, begann ich, auf ein Modell der nächsten Generation zu sparen.** In meinem Leben war das eine Riesensache. Wie gesagt, mir ist entfallen, wer in meiner Familie wann wo lebte, und ich weiß auch eine Menge anderer Dinge nicht mehr, aber den Weg zu meinem zweiten Computer werde ich so leicht nicht vergessen.

Ich hatte etwas Weihnachts- und-Geburtstags-Geld zurückgelegt (da ich am 28. Dezember geboren bin, mache ich zwischen den beiden Festen keinen großen Unterschied). Außerdem hatte ich einmal im Sommer ein bisschen Geld verdient, als ich in dem für die Parks von Helsinki zuständigen Reinigungstrupp arbeitete. Viele der Parks von Helsinki sind weder landschaftlich gestaltet noch gut gepflegt, sondern naturbelassene Wälder, die als Erholungsgebiete und grüne Lungen dienen. Unsere Aufgabe war es, zu groß gewordene Sträucher abzusägen oder abgestorbene Äste aufzusammeln - und das war sogar ganz interessant. Ich war schon immer ein Naturliebhaber. Außerdem hatte ich eine Zeit lang Zeitungen ausgetragen, nur dass es keine Zeitungen waren, sondern Werbeblätter. Eigentlich hatte ich mit Ferienjobs nicht viel am Hut. Trotzdem nahm ich sie an. Aber insgesamt habe ich wahrscheinlich mehr Geld durch Stipendien bekommen.

In Finnland ist es relativ üblich, dass Leute Schulen mit Stiftungen unterstützen, sogar öffentliche Grundschulen. Deshalb wird ab der vierten Klasse Geld an Schüler verteilt, je nach Maßgabe des Stifters. Ich erinnere mich, dass es bei uns unter anderem eine Stiftung für das beliebteste Kind in der Klasse gab. Ich war damals in der Sechsten, und wir wählten tatsächlich, welches Kind in der Klasse das Geld bekommen sollte. Vielleicht sollte ich hinzufügen, dass ich den Preis nicht gewann. Der Geldsegen beschränkte sich auf etwa 200 Finnmark, was damals vielleicht 40 Dollar entsprach, aber ich fand, es war viel Geld für einen Sechstklässler, der dafür lediglich beliebt zu sein brauchte.

Ziemlich häufig ging das Geld an den besten Schüler in einem Fach oder in einer Sportart. Und viele der Preise waren schulspezifisch oder regierungsfinanziert. In einigen Fällen war das Geld im Lauf der Zeit ausgegangen. Ich erinnere mich an einen Preis, der nur noch einen Wert von etwa einem Penny hatte. In solchen Situationen sprang die Schule mit einem Zuschuss ein, aber die Geldsumme blieb trotzdem ziemlich klein. Es ging vor allem um die Aufrechterhaltung der Tradition, jedes Jahr Geld zu verteilen. Finnland nimmt seine akademischen Traditionen ernst, und das ist gut so.

Auf diese Weise erhielt ich jedes Jahr irgendein Stipendium, weil ich „Der Mathetyp“ war. Als ich in die High-School kam, wurden die Preise größer. Die größten beliefen sich auf etwa 500 Dollar.

ter; mein wöchentliches Taschengeld hätte dafür niemals ausgereicht. Außerdem lieh ich mir etwas Geld von meinem Vater. Es war 1986 oder 1987. Ich war sechzehn oder siebzehn. Meine Basketballjahre lagen hinter mir. Ich verbrachte Unmengen von Zeit damit, das Feld zu sondieren, bevor ich mich entschied, was ich kaufen wollte. Die PCs waren damals noch nicht sehr gut, und wenn ich von meiner neuen Kiste träumte, wusste ich, dass sie kein PC sein würde.

Ich entschied mich für einen Sinclair QL, und viele von Ihnen sind wahrscheinlich zu jung, um ihn noch zu kennen. Hier ist seine Geschichte. Der Sinclair war eine der ersten 32-Bit-Maschinen auf dem Heimcomputermarkt. Sir Clive Sinclair, der Gründer der Firma, war der Steve Wosniak Großbritanniens. Er stellte diese Computerbausätze her, die in den USA als Timex-Computer verkauft wurden. Genau, die gleiche Firma, die die Uhren herstellte, importierte die Sinclair-Computerteile und verkaufte sie hierzulande unter dem Namen Timex. Die ersten wurden als Bausätze angeboten, und dann fing Sinclair an, fertig montierte Computer zu verkaufen.

Der Sinclair hatte ein Betriebssystem namens Q-DOS. Ich kannte es damals in- und auswendig. Es war eigens für diesen speziellen Computer geschrieben worden. Es hatte ein für die damalige Zeit ziemlich hochentwickeltes BASIC und besaß eine relativ brauchbare Grafik. Mich faszinierte unter anderem seine Multitasking-Fähigkeit. Man konnte mehrere Programme gleichzeitig laufen lassen. Allerdings war der BASIC-Teil nicht Multitasking-fähig, so dass man nicht mehr als ein BASIC-Programm auf einmal ausführen konnte. Aber wenn man seine eigenen Programme in Assemblersprache schrieb, konnte man das Betriebssystem veranlassen, sie zu schedulen und in Zeitscheiben aufzuteilen, und auf diese Weise konnte man viele von ihnen gleichzeitig ausführen. Der Computer enthielt den 8 Megahertz 68008-Chip, der die zweite und billigere Version des 68000-Chip von Motorola war. Intern waren die 68000-Chips der ersten Generation 32 Bit breit, aber extern hatten sie eine 16 Bit breite Schnittstelle zu allem, was außerhalb der CPU war - zum Beispiel zum Speicher oder zu zusätzlicher Hardware. Weil die CPU jeweils nur 16 Bit aus dem Speicher laden konnte, waren die 16-Bit-Operationen oft schneller als die 32-Bit-Operationen. Die Architektur war extrem populär und ist noch heute in vielen Embed-

ded-Systemen oder in Autos zu finden. Der Chip hat sich verändert, aber die Architektur ist im Prinzip die gleiche geblieben.

Der 68008-Chip, die Version in meinem Computer, verwendete 8 Bit und nicht 16 Bit als Schnittstelle zur Welt außerhalb der CPU. Aber obwohl er bei Interaktionen mit der Außenwelt immer nur acht Bit auf einmal verarbeiten konnte, war er intern ein 32-Bit-Prozessor. Dadurch ließ er sich in vieler Hinsicht angenehmer programmieren.

Er hatte 128 Kilobyte Speicher - keine Megabytes -, und das war damals für einen Heimcomputer gigantisch. Mein VIC-20, den er ersetzte, hatte nur 31/2 Kilobyte Speicher. Und weil er eine 32-Bit-Maschine war, konnte er ohne das geringste Problem auf den gesamten Speicher zugreifen - eine beispiellose Leistung für damalige Verhältnisse. Das war der Hauptgrund, warum ich den Computer kaufen wollte. Die Technologie war interessant. Und die CPU einfach genial.

Eigentlich hatte ich gehofft, Rabatt für meinen Computer zu bekommen, wenn ich ihn in einem Geschäft kaufte, mit dessen Inhaber ein Freund von mir bekannt war. Aber die Lieferung des Computers hätte so lange gedauert, dass ich einfach zu Akademiska Bokhandeln hinunter ging, der größten Buchhandlung in Helsinki, die eine Computerabteilung hatte. Dort konnte ich ihn gleich mitnehmen.

Der Computer kostete fast 2000 Dollar. Früher gab es diese Regel, dass Einstiegscomputer immer 2000 Dollar kosteten. Erst in den letzten paar Jahren hat sich das geändert. Jetzt können Sie einen neuen PC für 500 Dollar kaufen. Es ist wie bei Autos. Niemand stellt ein Auto für unter 10000 Dollar her. Irgendwann ist der Punkt erreicht, wo es sich einfach nicht mehr lohnt. Klar, die Firmen können ein Auto herstellen, das für 7000 Dollar verkauft werden kann, aber die Autohersteller sagen sich, dass die Leute, die sich ein Auto für 7000 Dollar leisten können, lieber eines für 10000 Dollar kaufen, das standardmäßig Zusatzausstattung wie eine Klimaanlage bietet. Wenn Sie heutige Einstiegsmodelle mit Einstiegsmodellen von vor 15 Jahren vergleichen, kosten sie etwa das Gleiche. Inflationsbereinigt kosten sie möglicherweise sogar etwas weniger. Aber sie sind viel besser.

So war es auch mit Computern. Als noch nicht jeder einen Computer kaufte, lag die Schmerzgrenze bei etwa 2000 Dollar. War der preiswerteste Computer deutlich teurer, konnte ein Unternehmen nicht sehr viele davon absetzen. Andererseits waren Computer in der Herstellung so teuer, dass es für ein Unternehmen nicht sinnvoll war, sie sehr viel

billiger anzubieten. Die meisten Leute waren bereit, 200 Dollar zusätzlich zu zahlen, wenn sie dafür ein besseres Gerät bekamen.

In den letzten beiden Jahren sanken die Herstellungskosten für Computer erheblich. Und selbst die Low-End-Kisten wurden ziemlich gut. Die Unternehmen verloren viele der Kunden, die für ein etwas besseres Gerät bereitwillig 200 Dollar zusätzlich zahlten. Und weil die Unternehmen die Käufer nicht mehr allein durch die Produkteigenschaften überzeugen konnten, mussten sie dazu übergehen, durch den Preis zu überzeugen.

Ich gebe es zu: 1987 war einer der Gründe, die für den Kauf des QL sprachen, sein cooles *Aussehen*.

Er war mattschwarz und hatte eine schwarze Tastatur. Er war ziemlich kantig. Das war keine wirklich schöne Kiste für Lackaffen. Der QL versuchte, irgendwie extrem zu wirken. Die Tastatur war an die drei Zentimeter dick, weil sie speziell zu diesem Computer gehörte. Das war bei den meisten Heimcomputern so. Auf der rechten Seite der Tastatur, wo normalerweise der Keypad war, gab es zwei Slots für das revolutionäre Sinclair Microdrive, das Endlosband, das nur die Sinclair-Maschinen hatten. Es erfüllte die gleichen Funktionen wie ein Diskettenlaufwerk und war auch so organisiert. Weil es ein langes Endlosband war, konntest du es einfach so weit rotieren lassen, bis du die gewünschte Stelle gefunden hattest. Es erwies sich als schlechte Idee, weil es nicht so zuverlässig arbeitete wie ein Diskettenlaufwerk.

So gab ich für den Sinclair QL fast 2000 Dollar aus. Er diente mir in der Hauptsache dazu, ein Programmierprojekt nach dem anderen durchzuführen. Ich war ständig auf der Suche nach interessanten Herausforderungen. Ich hatte einen Forth-Interpreter und -Compiler, nur um damit herumzuspielen. Forth war eine seltsame Sprache, die heute kein Mensch mehr benutzt. Sie war eine witzige Sprache für einen Nischenmarkt, die in den 1980er-Jahren für ganz unterschiedliche

Zwecke eingesetzt wurde, sich aber nie sonderlicher Beliebtheit erfreute. Leute, die keine Technik-Freaks waren, konnten sie kaum verstehen. Eigentlich war sie ziemlich nutzlos.

Ich schrieb mir meine eigenen Programmierertools. Zu den ersten Dingen, die ich für den Rechner kaufte, gehörte ein Erweiterungssteckplatz mit einer EEPROM-Karte (Electrically Erasable and Programmable Read Only Memory). Das ist ein Speicher, der mit einem Spezialmodul beschrieben wird und seine Daten nicht verliert, wenn

man den Computer abschaltet. Auf diese Weise konnte ich beliebig oft auf die Tools zugreifen, ohne sie in den RAM-Speicher laden und wertvollen RAM-Speicher für Programme opfern zu müssen.

Ich begann mich für Betriebssysteme zu interessieren, als ich einen Floppy-Controller kaufte, um nicht mehr auf die Microdrives angewiesen zu sein. Aber der Treiber für den Floppy-Controller war so schlecht, dass ich mir schließlich selbst einen schrieb. Bei dieser Gelegenheit fand ich ein paar Fehler im Betriebssystem - oder jedenfalls eine Diskrepanz zwischen dem, was das Betriebssystem laut Dokumentation tun sollte, und dem, was es tatsächlich tat. Die Unstimmigkeit fiel mir auf, weil etwas, was ich geschrieben hatte, nicht funktionierte.

Mein Code ist immer, ähm, perfekt. Deshalb wusste ich, dass etwas anderes der Grund sein musste, und machte mich daran, das Betriebssystem zu disassemblieren.

Du kannst dir Bücher kaufen, die Auszüge der Betriebssystemlistings enthalten. Das hilft. Außerdem brauchst du einen Disassembler, ein Werkzeug, das die Maschinensprache nimmt und in Assemblersprache zurückverwandelt. Das ist wichtig, denn wenn du nur eine Version in Maschinensprache hast, ist es schwierig, die Befehle zu verfolgen. Du stellst fest, dass ein Befehl zu einer numerischen Adresse verzweigt und sich daher nur sehr schwer lesen lässt. Ein guter Disassembler gibt den Zahlen Namen und erlaubt es dir, selbst Namen zu vergeben. Du kannst ihn auch dazu verwenden, dir bei der Identifizierung bestimmter Befehlssequenzen helfen zu lassen. Ich hatte meinen eigenen Disassembler, mit dessen Hilfe ich ziemlich schöne Listings erzeugen konnte. Wenn etwas nicht funktionierte, konnte ich ihn nach dem Listing für eine bestimmte Stelle suchen lassen, und ich konnte alles sehen, was das Betriebssystem tat. Manchmal verwendete ich den Disassembler nicht für die Fehlersuche, sondern um zu verstehen, wie etwas funktionierte.

Eines der Dinge, die ich am QL hasste, war sein Read-only-Betriebssystem. Man konnte nichts verändern. Es hatte einige Hooks. Hooks sind Stellen, wo du deinen eigenen Code einfügen kannst, wenn er bestimmte Funktionen übernehmen soll. Aber Hooks gab es nur an bestimmten Stellen. Es ist viel schöner, wenn du dein Betriebssystem komplett ersetzen kannst. Es ist eine schlechte Idee, ein Betriebssystem ins ROM (Nur-Lese-Speicher) zu stecken.

Obwohl ich Finnland als ein Land beschrieben habe, das neue Technologien energisch vorantreibt, konnte der Sinclair QL sich in Europas siebtgrößter Nation nicht richtig durchsetzen. Weil der Markt so klein war, musstest du jeden einzelnen Upgrade für deine bilderstürmerische, bahnbrechende Maschine per Post aus England beziehen. Das bedeutete, Kataloge zu wälzen, bis du jemanden fandest, der das Gewünschte verkaufte. Dann musstest du dich um Schecks kümmern, sie als gedeckt bestätigen lassen und wochenlang auf die Lieferung warten. (Das war in den Zeiten vor Kreditkarten und Amazon.) Dieses Trara war notwendig, um meinen RAM-Speicher von 128 Kilobyte auf 640 Kilobyte zu erweitern. Dieser Übung unterzog ich mich, als ich einen neuen Assembler für die Übersetzung von Assemblersprache in Maschinencode (die Einsen und Nullen) und einen Editor kaufte, eine Art Textverarbeitungsprogramm, mit dem man Programme schreibt.

Sowohl der neue Assembler als auch der Editor funktionierten gut, aber sie befanden sich auf den Microdrives und konnten nicht in das EEPROM geladen werden. Deshalb schrieb ich mir selbst einen Editor und einen Assembler und nutzte sie, um zu programmieren. Beide waren in Assemblersprache geschrieben, was gemessen an heutigen Standards extrem unintelligent ist. Es ist kompliziert und zeitaufwändig - über den Daumen gepeilt dauert es mit Assembler vielleicht hundert Mal so lang, für ein Problem eine Lösung zu finden, wie mit der Sprache C, die es damals bereits gab.

Ich fügte dem mitgelieferten Interpreter des Rechners ein paar Befehle hinzu, so dass ich, wenn ich etwas editieren wollte, im Prinzip nur meinen Editor aufrufen musste und loslegen konnte. Mein Editor war schneller als der Editor, der mit dem Rechner mitgeliefert worden war. Ich war besonders stolz darauf, wie schnell er Zeichen auf dem Bildschirm ausgeben konnte. Normalerweise dauerte es mit einem Rechner wie diesem so lange, den Bildschirm mit Zeichen zu füllen, dass man sehen konnte, wie der Text scrollte. Deshalb war ich höchst angetan von der Tatsache, dass man mit meinem Editor Text so schnell schrieb, dass man beim schnellen Scrollen eine Art Schleier vorbeiziehen sah. Das war mir wichtig. Es sah besser aus, und ich wusste, dass ich eine Menge getan hatte, um diesen schnellen Textaufbau zu erreichen.

Zu dieser Zeit kannte ich nicht sehr viele Leute, die sich so eingehend mit Computern beschäftigten wie ich. Es gab einen

der Schule, aber dort war ich nicht oft. Er war im Prinzip für Kids gedacht, die etwas über Computer erfahren wollten. In meiner High-School gab es insgesamt nur 250 Schüler, und ich glaube nicht, dass noch ein anderer Schüler so wie ich einen Computer benutzt hatte, seit er zehn war.

Besonders gern nutzte ich den Sinclair QL, um Spiele zu klonen. Ich schrieb Klone der Spiele des VIC-20, die ich am liebsten gespielt hatte. Manchmal erweiterte ich sie. Aber meistens wurden sie dadurch nicht besser. Bessere Kiste, unverändertes Konzept.

Mein Lieblingsspiel war vermutlich Asteroids, aber es gelang mir nie, einen guten Klon davon zu machen. Das lag daran, dass damals alle Asteroids, die auf Spielekonsolen liefen, mit echter Vektorgrafik realisiert waren. Statt einer Grafik, die auf kleinen Punkten - Pixeln - basierte, besaßen sie eine Grafik, die nach dem gleichen Prinzip funktionierte wie die Kathodenstrahlröhre, das heißt, aus einer Elektronenkanone werden Elektronen von hinter der Kathodenstrahlröhre herausgeschossen und mit Magneten abgelenkt. Auf diese Weise erhielt man eine viel höher auflösende Grafik, die sich aber nicht so ohne weiteres nachahmen ließ. Man konnte einen Klon programmieren, aber er sah nicht wie die Original-Version von Asteroids aus, wenn man ihn auf einem Computer schrieb, der diese besondere Grafikfähigkeit nicht besaß.

Ich erinnere mich daran, dass ich einen Pacman-Klon in Assembler programmierte. Dazu rufst du dir als Erstes vor Augen, wie die Pacman-Figuren ungefähr aussehen. Dann zeichnest du sie in Farbe in einem 16 x 16-Raster auf Papier. Und wenn du künstlerisch begabt bist, gelingt dir das ganz gut. Aber wenn du wie ich Mr. KünstlerischUnbegabt bist, sieht deine Figur wie Pacmans kranker Cousin aus.

Okay, der Klon war nicht besonders gut. Aber ich war sehr stolz darauf. Man konnte das Spiel wirklich spielen, und ich sandte es an eines der Magazine, die Computercode veröffentlichten. Ich hatte schon andere Programme an Computermagazine gesandt und dachte, dieses würde sich bestens dafür eignen.

Tat es aber nicht.

Eines der Probleme bestand darin, dass das Programm in Assemblersprache geschrieben war. Das bedeutete, dass es nicht funktionierte, wenn dir beim Abschreiben aus dem Magazin auch nur der winzigste, *winzigste* Fehler unterlief.

Ich schrieb auch ein paar Spiele, die ich mir selbst ausgedacht hatte. Aber man braucht eine bestimmte Art zu denken, um Spiele zu ersinnen. Weil Spiele sehr viel Rechnerleistung benötigen, musst du wirklich tief in die Hardware des Computers einsteigen. Das konnte ich. Aber ich hatte nicht die Mentalität eines Spielers. Wie toll ein Spiel ist, entscheidet sich normalerweise weder an seiner Schnelligkeit noch an seiner Grafikqualität. Es muss etwas geben, was dich dazu bringt, es zu spielen - etwas, was dich bei der Stange hält. Es ist genau wie bei Filmen. Spezialeffekte sind das eine. Aber du brauchst auch einen Plot. Und meine Spiele hatten nie einen Plot. Ein Spiel benötigt eine Idee, eine Steigerung. Häufig besteht die Steigerung nur darin, dass das Spiel schneller wird. Wie bei Pacman. Manchmal verändert sich das Labyrinth, oder die Monster werden bei der Verfolgung schlauer.

An Pacman interessierte mich unter anderem, wie man es anstellen konnte, dass die Grafik nicht flimmert. Dieses Problem tritt bei älteren Computerspielen häufig auf: Ohne Spezialhardware ist ein Flimmern der Figuren unvermeidlich. Du bewegst sie, indem du das alte Männchen entfernst und ein neues Männchen ausgibst. Ist dein Timing schlecht, können die Leute sehen, wenn kein Männchen da ist, und deshalb flimmert es. Es gibt verschiedene Möglichkeiten, mit diesem Problem umzugehen. Du kannst zum Beispiel zuerst die neue Figur ausgeben und erst *danach* die alte entfernen. Aber du musst aufpassen, dass du nicht den Teil des alten Männchens entfernst, der durch die neue Figur überdeckt wird. Anstelle eines irritierenden Flackerns erhältst du einen schönen Effekt - du siehst manchmal den Schatten der alten Figur auf dem Bildschirm. Das Gehirn interpretiert das auf seine Weise - nicht als Flackern, sondern als verschwommene Bewegung. Allerdings ist diese Lösung in der Realisierung ziemlich teuer und zeitaufwändig.

Es hat seinen Grund, warum Spiele sich immer an der Grenze des gerade noch Machbaren bewegen und warum Programmierer sich bevorzugt als Erstes diesem Programmtyp zuwenden. Zum Teil liegt es daran, dass ein paar der cleversten Programmierer dort draußen 15-jährige Jungs sind, die zu Hause in ihrem Zimmer an ihrem Com-

puter herumbasteln. (Das habe ich schon vor sechzehn Jahren gedacht, und davon rücke ich auch heute nicht ab.) Aber Spiele haben noch aus einem anderen Grund eine Vorreiterfunktion: Spiele reizen die Hardware oft bis zum Äußersten aus.

Heutige Computer sind normalerweise für alle Anwendungen schnell genug. Bei Action-Spielen aber, zum Beispiel den beliebten 3-D-Spielen, stößt die Hardware wirklich an ihre Grenzen. Spiele gehören zu den wenigen Computerprogrammen, bei denen es auffällt, wenn die Dinge nicht in Echtzeit passieren. Bei einem Textverarbeitungsprogramm kommt es auf eine Verzögerung um eine Sekunde hier oder eine Sekunde da nicht an. In einem Spiel aber fallen Verzögerungen

bereits auf, wenn sie weniger als eine Zehntel Sekunde dauern. Die Spiele von damals waren vom Prinzip her ziemlich einfach gestrickt. Heute macht die Programmierung nur einen ziemlich kleinen Teil der Spiele aus. Was wirklich zählt, sind Musik und Handlung. Vergleicht man ein Computerspiel mit einem Film, ist die Programmierkomponente lediglich die Kameraarbeit.

Ich hatte den Sinclair QL drei Jahre lang. Er begleitete mich von der High-School an die Universität Helsinki und sogar noch in die finnische Armee. Er war okay, aber wir hatten uns definitiv auseinander gelebt. Im letzten Jahr oder so hatte ich seine Unzulänglichkeiten aufgedeckt. Der 68008 war als CPU ganz gut, aber ich las über die nächste Generation, den 68020, und lernte Vorzüge wie Speicherverwaltung und Paging kennen. Die neuen Computer konnten Sachen, die wirklich wichtig sind, wenn man sich mit Low-Level-Fragen befasst.

Am Sinclair QL wurmte mich besonders, dass das Betriebssystem trotz seiner Multitasking-Fähigkeit jederzeit abstürzen konnte, weil es keinen Speicherschutz gab. Wenn ein Task Mist baute, stürzte die Kiste einfach ab.

Der Sinclair QL war Sir Clive Sinclairs letzter Versuch, Computer zu entwickeln und herzustellen. Das lag unter anderem daran, dass der QL kommerziell nicht erfolgreich war. Er hatte eine interessante Technologie, aber die Firma hatte Produktionsprobleme und Schwierigkeiten mit der Qualitätssicherung und die unvermeidliche schlechte Presse. Darüber hinaus nahm die Konkurrenz am Markt zu.

In den späten 1980er-Jahren konnte man sich allmählich vorstellen, dass eines Tages vielleicht der ganz normale Fahrgast in der Straßenbahn einen Computer haben würde, und sei es nur, um Textverarbeitung zu machen. Und alles deutete auf den PC hin. Ja, die Regale begannen, sich unter den Original-IBM-PCs zu biegen, deren Erfolg trotz zahlreicher technischer Unzulänglichkeiten nicht aufzuhalten

war. Schließlich trugen diese allgegenwärtigen beigebraunen Geschöpfe das Gütezeichen IBM, und das bedeutete viel. Eine weitere Attraktion: Die Peripheriegeräte waren standardisiert und leicht zu beschaffen.

Ich las von all diesen neuen CPUs, die genau das *konnten*, was ich wollte. Es wurde mir klar, dass der 68020, so interessant er auch *aussehen* mochte, keine Zukunft haben würde. Eine Möglichkeit wäre es gewesen, einen CPU-Upgrade für den QL zu kaufen. In jenen Tagen bedeutete das praktisch einen Komplettumbau der Kiste. Und da Speicherverwaltung für das Betriebssystem ein Fremdwort war, hätte ich mir selbst eine schreiben müssen. Die Situation war also folgende: *Hhhmmm. Das wäre ein Riesenaufwand. Und der Kauf einer neuen CPU wäre ganz schön teuer.*

Dazu kam, dass mich der Kauf von Computerbauteilen zunehmend nervte. Es gab für den Sinclair QL ja keinen Sears-Katalog, der es mir erlaubt hätte, zum Telefon zu greifen und mehr Speicher zu ordern. Die ewigen Postbestellungen aus England hatte ich satt. (Dagegen machte es mir nichts aus, dass es keine in Folie eingeschweißte Software gab, denn das alles konnte ich mir selbst schreiben.)

Diese quälende Ungewissheit hatte eine positive Nebenwirkung. Während ich darüber nachdachte, die Kiste abzustößen, beschloss ich, meine Peripherie zu verkaufen - das echte Plattenlaufwerk, das ich gekauft hatte, weil ich das Microdrive keine Sekunde länger ertragen konnte, und meinen Erweiterungs-RAM-Speicher. Weil die Leute, die an solchen Sachen Interesse hatten, nicht draußen auf der Straße Schlange standen, blieb mir nichts anderes übrig, als in einer Computerzeitschrift zu inserieren und zu beten. Und so lernte ich meinen guten Freund Jouko Vierumaki kennen, der, wie sich herausstellte, wahrscheinlich der einzige Mensch in ganz Finnland war, der außer mir einen Sinclair QL besaß. Er antwortete auf meine Anzeige, kam mit dem Zug aus Lahti und kaufte mir einen Teil meiner Peripherie ab. Dann brachte er mir Billard bei.

II.

In meinem ersten Jahr an der Uni stand der Sinclair QL zwar auf einem Schreibtisch vor dem Fenster meines Schlafzimmers im ersten Stock in der Petersgatan, aber ich programmierte nur selten. Das lag zum Teil daran, dass ich mich auf mein Studium konzentrieren wollte. Vor allem aber fehlte mir ein Projekt, das mich fesselte. Und wenn dir das richtige Projekt fehlt, fehlt dir auch der Schwung. Also versuchst du, etwas anderes zu finden, was dich motivieren könnte.

Es schien mir der perfekte Moment zu sein, meinen Wehrdienst abzuleisten, der mir, wie ich wusste, ohnehin bevorstand. Ich war 19 Jahre alt, von der Unvollkommenheit meines Computers angeenervt und durch kein interessantes Computerprojekt gebunden. Ich bestieg einen Zug nach Lappland.

Ich habe schon erwähnt, dass ich keine Ahnung hatte, welche körperlichen Anforderungen der Wehrdienst stellen würde. Nachdem ich elf Monate Sportunterricht-an-der-Waffe hinter mich gebracht hatte, empfand ich es deshalb als mein gutes Recht, die verbleibenden Jahrzehnte meines Lebens in himmlischer Bewegungslosigkeit zu genießen und meine sportlichen Aktivitäten darauf zu beschränken, Code auf einer Tastatur einzugeben oder meine Finger um ein Pils-Glas zu legen. (Tatsächlich fand meine erste sportähnliche Aktivität nach dem Verlassen der Armee fast auf den Tag genau zehn Jahre nach meiner Entlassung statt, als David mich dazu brachte, mit ihm zum Boogie-Boarden zu gehen und die Killerwellen von Half Moon Bay hinunter zu surfen. Ich bin dabei fast ertrunken, und meine Beine taten mir tagelang weh.)

Mein Wehrdienst endete am 7. Mai 1990. Während Tove Ihnen sagen könnte, dass ich mich nur mit größter Mühe an an das Datum

Hochzeitstages erinnere, ist es mir unmöglich, das Datum meiner Entlassung aus der Armee zu vergessen.

Als Erstes wollte ich mir eine Katze anschaffen.

Ich hatte einen Freund, dessen Katze vor ein paar Wochen geworfen hatte, und kaufte kurzerhand das einzige Junge, das noch übrig war, einen schönen kleinen weißen Kater, der sich, weil er seine ersten Lebenswochen im Freien verbracht hatte, problemlos innerhalb und außerhalb der Wohnung meiner Mutter zurecht fand. Ich nannte ihn Randi, als Abkürzung von Mithrandir, dem weißen Zauberer in *Herr der Ringe*. Er ist jetzt elf Jahre alt und wie sein Besitzer völlig an den kalifornischen Lebensstil angepasst.

Nein, ich glaube nicht, dass ich in jenem Sommer etwas Produktives geleistet habe. Die Vorlesungen für mein zweites Jahr an der Uni würden erst im Herbst wieder beginnen. Mein Computer war nicht mehr auf der Höhe der Zeit. Deshalb hing ich in meinem heruntergekommenen Bademantel herum, spielte mit Randi oder traf mich gelegentlich mit Freunden und gab ihnen Gelegenheit, über meine Bowling- oder

Billard-Versuche zu lästern. Okay, ein bisschen träumte ich auch von meinem nächsten Computer.

Ich befand mich in einem typischen Geek-Dilemma. Wie jeder echte Computer-Purist, der mit einem 68008-Chip aufgewachsen war, verachtete ich PCs. Aber als 1986 der 386er auf den Markt kam, begannen PCs, ähm, attraktiv zu werden. Sie konnten alles, was der 68020 konnte, und wurden Ende der 1980er, Anfang der 1990er-Jahre durch die Massenproduktion und die Einführung preiswerter IBM-kompatibler Rechner sehr viel billiger. Ich ging mit Geld sehr vorsichtig um, weil ich keines hatte. Das gab den Ausschlag: Die Kiste wollte ich haben. Und weil PCs florierten, würde ich Updates und Zubehör problemlos bekommen können. Im Hardwarebereich wollte ich etwas haben, was dem Standard entsprach.

Ich beschloss, überzulaufen und die Fronten zu wechseln. Es würde Spaß machen, eine neue CPU zu beschaffen. Stück für Stück verscherbelte ich die Bauteile meines Sinclair QL.

Jeder hat ein Buch, das sein oder ihr Leben verändert hat. *Die Heilige Schrift. Das Kapital. Dienstags mit Morrie. Alles was ich wissen musste, lernte ich im Kindergarten.* Was auch immer. (Ich hoffe ehrlich, dass für Sie, nachdem Sie das Vorwort und meine Theorie über den Sinn des Lebens gelesen haben, das vorliegende Buch diesen Platz

in Ihrem Leben einnehmen wird.) Das Buch, das mir neue Welten erschloss, war *Operating Systems: Design and Implementation* von Andrew S. Tanenbaum.

Ich hatte mich bereits für meine Kurse im Herbst angemeldet und war am meisten auf die Vorlesung über die Programmiersprache C und das Betriebssystem Unix gespannt. Als Vorbereitung darauf und um mir einen Wissensvorsprung zu verschaffen, kaufte ich mir im Sommer das oben erwähnte Buch. Andrew Tanenbaum, ein Universitätsprofessor in Amsterdam, diskutiert darin Minix, eine für Lernzwecke entwickelte Unix-Version. Minix ist eine Art einfacher Unix-Klon. Nachdem ich die Einleitung gelesen und die Philosophie hinter Unix und die Leistungsmöglichkeiten eines mächtigen, sauberen und schönen Betriebssystems kennen gelernt hatte, beschloss ich sehr schnell, dass ich einen Rechner wollte, auf dem man Unix ausführen konnte. Ich würde Minix darauf fahren, die einzige einigermaßen nützliche Unix-Version, die ich finden konnte.

Als ich weiterlas und anfang, Unix zu verstehen, durchströmte mich ein unglaublicher Enthusiasmus. Ehrlich gesagt, er beflügelte mich bis heute. (Ich hoffe, in Ihrem Leben gibt es etwas, wovon Sie das Gleiche behaupten können.)

III.

Das akademische Jahr, das im Herbst 1990 begann, war das erste Jahr, in dem es an der Universität Helsinki Unix gab, das mächtige Betriebssystem, das Ende der 1960er-Jahre in den Bell Labs von AT&T geboren wurde, aber anderswo erwachsen geworden war. In meinem ersten Studienjahr hatten wir noch eine VAX gehabt, die unter VMS lief. Das war ein grauenhaftes Betriebssystem und gewiss keine Umgebung, von der du dir gewünscht hättest: „Wow, das hätte ich auch gern zu Hause.“ Stattdessen ließ es dich stöhnen: „HmMMM. Wie soll *das* denn gehen?“ Es war schwer zu benutzen. Es besaß nur wenige Tools. Es erlaubte keinen problemlosen Zugriff auf das Internet, das unter Unix lief. Man konnte noch nicht einmal mühelos herausfinden, wie groß eine Datei war. Zugegebenermaßen eignete sich VMS sehr gut für bestimmte Aufgaben wie Datenbanken. Aber es ist nicht die Art von Betriebssystem, die einen begeistern kann.

Die Universität erkannte, dass es an der Zeit war, davon wegzukommen. Ein Großteil der akademischen Welt war damals von Unix zunehmend angetan, und so schaffte die Universität eine MicroVAX an, auf der Ultrix lief, die Unix-Version von Digital Equipment. Es war eine Möglichkeit, in Unix einzusteigen.

Ich brannte darauf, mit Unix zu arbeiten und mein neuerworbenes Wissen aus Andrew Tanenbaums Buch anzuwenden, und stellte mir begeistert vor, was ich damit alles anfangen könnte, wenn ich einen 386er-PC hätte. Ich sah keine Möglichkeit, die 18000 FM zusammenzubekommen, die er kosten sollte. Aber ich wusste, dass ich ab Beginn des Herbstsemesters über meinen Sinclair QL auf den neuen UnixComputer der Universität würde zugreifen können, bis ich mir einen PC leisten konnte, auf dem ich selbst Unix nutzen konnte.

So tat ich also in diesem Sommer zweierlei: Nichts. Und ich verschlang die 719 Seiten von *Operating Systems: Design and Implementation*. Das rote Flexcover-Buch lebte gewissermaßen auf meinem Bett.

Die Universität Helsinki ließ für die MicroVAX eine Lizenz für sechzehn Anwender springen. Das bedeutete, dass die Teilnehmerzahl an der „C und UNIX“-Veranstaltung auf zweiunddreißig Studierende beschränkt war - ich nehme an, man stellte sich vor, sechzehn Leute würden tagsüber und sechzehn nachts damit arbeiten. Wie für uns alle war Unix auch für den Dozenten ein völlig unerforschtes Gebiet. Das gab er offen zu, und damit war die Sache kein Problem. Aber er war dem Unterrichtsstoff jeweils nur ein Kapitel im Buch voraus, während die Studenten manchmal drei Kapitel weiter sprangen. Deshalb machten wir eine Art Spiel daraus, dem Dozenten durch Fragen über Themen, die erst drei Kapitel später drankamen, ein Bein zu stellen, um zu sehen, ob er schon so weit gekommen war.

Wir alle betraten mit Unix Neuland, und saßen in einem Kurs, der erst im Aufbau begriffen war. Aber der Kurs zeigte uns, dass hinter Unix eine einzigartige Philosophie stand. Das begriff man schon in der ersten Vorlesung. Der Rest bestand aus der Klärung von Details. Das Besondere an Unix sind die grundlegenden Ideale, die es zu verwirklichen sucht. Es ist ein sauberes und schönes Betriebssystem. Es vermeidet Spezialfälle. Unix baut auf dem Prozessbegriff auf - alles, was etwas tut, ist ein Prozess. Hier ist ein einfaches Beispiel. In Unix ist das Shell-Kommando, das du eingibst, um Zutritt zum Betriebssystem zu erlangen, anders als bei DOS nicht ins Betriebssystem eingebaut. Es ist nur ein Task. Wie jeder andere Task auch. Zufällig liest dieser Task von deiner Tastatur und schreibt deine Eingabe auf deinen Bildschirm. Alles, was in Unix etwas tut, ist ein Prozess. Daneben gibt es Dateien.

Dieses simple Design ist es, was mich faszinierte, und was die meisten Leute an Unix fasziniert (oder jedenfalls uns Geeks). Fast alles, was du in Unix machst, wird von gerade mal sechs elementaren Operationen erledigt (so genannten *System calls*, Systemaufrufen, mit denen du das Betriebssystem aufforderst, etwas für dich zu erledigen.) Aus diesen elementaren Systemaufrufen kannst du praktisch alles andere zusammenbauen.

Dann gibt es das Konzept des *fork*, das zu den grundlegenden Unix-Operationen gehört. Wenn ein Prozess einen *fork* ausführt, legt er eine komplette Kopie seiner selbst an. Auf diese Weise bekommst du zwei

identische Kopien. Die Kind-Kopie wird meistens dazu benutzt, einen anderen Prozess *auszuführen* - *sich* selbst durch ein neues Programm zu ersetzen. Das ist die zweite elementare Operation. Darüber hinaus gibt es vier weitere elementare Systemaufrufe: *open*, *close*, *read* und *write* - die alle dafür entworfen sind, auf Dateien zuzugreifen. Aus diesen sechs Systemaufrufen setzen sich die einfachen Operationen zusammen, die das Rückgrat von Unix bilden.

Klar, es gibt Unmengen anderer Systemaufrufe, die all die Details erledigen. Aber sobald du die sechs elementaren Aufrufe verstanden hast, hast du Unix verstanden. Zum Charme von Unix gehört nämlich die Erkenntnis, dass du keine komplexen Schnittstellen brauchst, um etwas Komplexes zu bauen. Du kannst durch die Interaktionen einfacher Elemente beliebig viel Komplexität erzeugen. Um komplexe Problemlösungen zu realisieren, erzeugst du Kommunikationskanäle zwischen einfachen Prozessen (die in Unix-Sprache *Pipes* heißen).

Ein hässliches System hat spezielle Schnittstellen für alles, was du tun willst. Unix ist das genaue Gegenteil davon. Es gibt dir die Bausteine in die Hand, mit denen du alles machen kannst. Das ist das Charakteristikum eines sauberen Entwurfs.

Für Sprachen gilt das Gleiche. Die englische Sprache hat 26 Buchstaben, aus denen man alles aufbauen kann. Ihr Gegenstück ist das Chinesische, das jeden nur denkbaren Begriff mit einem eigenen Schriftzeichen darstellt. Die chinesischen Schriftzeichen sind hochkomplex, lassen sich aber nur begrenzt miteinander kombinieren. Das entspricht eher dem VMS-Ansatz: Komplexe Grundelemente haben interessante Bedeutungen, sind aber nicht variabel einsetzbar. Auch Windows verfolgt diesen Ansatz.

Unix dagegen beruht auf der Philosophie *Small is beautiful* - die Schönheit liegt im Kleinen. Eine kleine Menge einfacher elementarer Bausteine kann zu einer unendlichen Komplexität des Ausdrucks kombiniert werden.

In der Physik ist es übrigens genauso. Man sucht nach fundamentalen Regeln, die relativ einfach sein sollen. Die Komplexität ergibt sich aus den unglaublichen Interaktionen, die zwischen diesen einfachen Regeln möglich sind, nicht aus der inhärenten Komplexität der Regeln selbst.

Diese Einfachheit kommt nicht von ungefähr. Das einfache Bausteinkonzept von Unix wurde von seinen Erfindern Dennis Ritchie und Ken Thompson in den Bell Labs von AT&T mit Bedacht geplant.

Glauben Sie nicht, Einfachheit sei leicht zu haben. Einfachheit erfordert *Design* und guten Geschmack.

Um auf das Beispiel der gesprochenen Sprachen

Bildhafte Schriften wie die Schriftzeichen der Chinesen oder Hieroglyphen entstehen in der Regel zuerst und sind „einfacher“, während der Bausteinansatz ein sehr viel abstrakteres Denken erfordert. Genauso wenig sollten Sie die Einfachheit von Unix mit mangelnder Raffinesse verwechseln - im Gegenteil.

Das heißt nicht, dass Unix aus erhabenen Beweggründen heraus entstand. Wie bei so vielen anderen Entwicklungen im Computerbereich war sein Nährboden der Spieltrieb. Es brauchte jemanden, der Computerspiele auf einer PDP-11 spielen wollte. Dafür nämlich wurde Unix zunächst entwickelt - als Dennis' und Kens persönliches Projekt, *Space Wars* zu spielen. Und weil das Betriebssystem nicht als ernsthaftes Projekt betrachtet wurde, galt es bei AT&T nicht als kommerzielle Unternehmung. Tatsächlich war AT&T eine regulierte Monopolgesellschaft und durfte ohnehin keine Computer verkaufen. Deshalb machten die Leute, die Unix entwickelten, es zusammen mit den Quelllizenzen relativ frei zugänglich, vor allem für die Hochschulen. Sie maßen ihm einfach keine große Bedeutung bei.

All das führte dazu, dass Unix im akademischen Bereich ein großes Projekt wurde. Als AT&T 1984 zerschlagen wurde und endlich ins Computergeschäft einsteigen durfte, hatten Informatiker an den Hochschulen - insbesondere an der University of California, Berkeley - unter der Leitung von Leuten wie Bill Joy und Marshall Kirk McKusik Unix seit Jahren verbessert und weiterentwickelt. Dabei hatten sie auf die Dokumentation ihrer Arbeit häufig wenig Wert gelegt.

Anfang der 1990er-Jahre aber hatte Unix sich zum gefragtesten Betriebssystem für Supercomputer und Server entwickelt. Es war zu einem Riesengeschäft geworden. Damit trat unter anderem das Problem auf, dass es mittlerweile eine Fülle konkurrierender Versionen des Betriebssystems gab. Einige waren aus den relativ kontrollierten Strukturen des AT&T-Codes hervorgegangen (den so genannten System-V Varianten), während andere vom Code der University of California, Berkeley, abgeleitet wurden (BSD - Berkeley Software Distribution). Wieder andere stellten eine Mischung aus beiden dar. Ein BSD-Derivat verdient besondere Erwähnung: das

Internet verteilte. Es wurde später in die frei zugänglichen Varianten NetBSD, FreeBSD und OpenBSD aufgesplittet, und fand viel Beachtung in der Unix-Gemeinde.

Das brachte AT&T auf den Plan. Das Unternehmen verklagte die University of California, Berkeley. Der ursprüngliche Code hatte AT&T gehört, aber der Großteil der nachfolgenden Weiterentwicklungen hatte in Berkeley stattgefunden. Die Hochschulleitung der University of California setzte dagegen, die Universität hätte das Recht, ihre Unix-Version gegen eine nominale Gebühr zu verteilen oder zu verkaufen. Und sie wies nach, dass die Universität so viel Arbeit in Unix gesteckt hatte, dass der von AT&T zugänglich gemachte Code praktisch neu geschrieben worden war. Der Prozess endete schließlich mit einem Vergleich, nachdem das Unternehmen Novell Unix von AT&T gekauft hatte. Im Prinzip mussten Teile des Systems entfernt werden, die aus dem von AT&T zugänglich gemachten Code stammten.

In der Zwischenzeit hatte das gerichtliche Hin und Her dazu gedient, einen neuen Stern am Horizont heranreifen und sich entfalten zu lassen. Es gab Linux die Zeit, sich am Markt zu etablieren. Aber ich greife mir vor.

Da ich ohnehin schon abschweife, möchte ich etwas erklären. Unix hat den Ruf, die Exzentriker am Rand der Computer-Gemeinde magnetisch anzuziehen. Dagegen lässt sich nichts einwenden. Es stimmt.

In der Tat interessieren sich eine Menge ziemlich - tja, wie soll ich es sagen - ziemlich verrückter Leute für Unix. Nicht verrückt in dem Sinn, dass sie sich wie hysterische Groupies gebärden. Nicht verrückt in dem Sinn, dass sie den Hund ihrer Nachbarn vergiften. Nur verrückt in dem Sinn, dass sie einen *sehr* alternativen Lebensstil pflegen.

Dazu muss man wissen, dass die ersten Unix-Aktivitäten vorwiegend in den späten 60er- und frühen 70er-Jahren stattfanden, während ich in einem Wäschekorb in der Wohnung meiner Großeltern schlummerte. Die ersten Unix-Anhänger gehörten der Flower-Power-Bewegung an - einer *technischen* Seite der Flower-Power-Bewegung. Die Unix-muss-frei-sein-Philosophie hat mehr mit der damaligen Lebenseinstellung als mit dem Betriebssystem zu tun. Es war eine Zeit wilder Ideale. Revolution. Nieder mit den Autoritäten. Freie Liebe. (Die ich versäumte und mit der ich ohnehin nicht viel hätte anfangen können.) Die relative Offenheit von Unix war zwar hauptsächlich dem fehlenden kommerziel-

len Interesse zur damaligen Zeit zu verdanken, aber sie ließ das Betriebssystem den Blumenkindern als attraktiv erscheinen.

Ich lernte diese Seite von Unix wahrscheinlich 1991 zum ersten Mal kennen, als Lars Wirzenius mich zu einem Vortrag an der Technischen Universität von Helsinki mitschleppte (die, wie jeder weiß, nicht wirklich in Helsinki liegt, sondern knapp jenseits der Stadtgrenze in Espoo. Man möchte dort gerne mit der angesehenen Universität von Helsinki assoziiert werden und sei es nur dem Namen nach). Der Redner war Richard Stallman.

Richard Stallman ist der Gott der freien Softwarebewegung. Er begann 1984 mit der Arbeit an einer Unix-Alternative, dem so genannten GNU-System. Der Name GNU steht für „GNU is Not Unix“, und gehört zu den vielen rekursiven Akronymen, bei denen einer der Buchstaben für das Akronym selbst steht - eine Art Insider-Witz, den außer Informatikern kein Mensch versteht. Wir Geeks sind einfach eine unheimlich witzige Gesellschaft.

Wichtiger aber: RMS, wie er sich gerne nennen lässt, schrieb auch das Free-Software-Manifest und die GPL, die General Public License für freie Software. Man kann sagen, er machte als Erster deutlich, dass das Konzept des frei zugänglichen Quellcodes einer bewussten Absicht entsprang und nicht nur einem Versehen, wie es bei der ursprünglichen offenen Unix-Entwicklung der Fall war.

Ich muss zugeben, dass mir die gesellschaftspolitischen Fragen, die RMS so am Herzen lagen - und liegen - kaum bewusst waren. Ich war mir auch der Free Software Foundation, die er gegründet hatte, und ihrer Anliegen nicht allzu bewusst. Aus der Tatsache, dass ich mich nur vage an RMS' Vortrag 1991 erinnere, schließe ich, dass er mich zu diesem Zeitpunkt nicht weiter beeindruckte. Mich interessierte die Technik, nicht die Politik - von Politik hatte ich zu Hause genug gehört. Aber Lars war Ideologe, und so kam ich mit und hörte mir die Sache an.

Mit Richard sah ich zum ersten Mal in meinem Leben den typischen langhaarigen, bärtigen Hackertyp. In Helsinki gibt es davon nicht allzu viele.

Sein Vortrag mag keine Erleuchtung für mich gewesen sein, aber etwas davon muss wohl hängen geblieben sein. Immerhin habe ich später die GPL für Linux genutzt. Verdammt, ich greife mir schon wieder vor.

IV.

z. Januar 1991.

Es war der erste Tag, an dem die Geschäfte nach Weihnachten und meinem 21. Geburtstag wieder geöffnet hatten - jenen beiden lukrativsten Tagen in meinem Kalender.

Mit meinem Weihnachts-und-Geburtstags-Geld in der Hand traf ich die gigantische Geschäftsentscheidung, einen Computer für 18 000 Finnmark zu kaufen. Das entsprach etwa 3500 Dollar. Weil ich so viel Geld nicht hatte, wollte ich ein Drittel des Kaufpreises anzahlen und den Computer auf Kredit finanzieren. Eigentlich kostete der Computer nur 15000 FM. Den Rest verschlangen die Kosten für den Kredit, den ich in den nächsten drei Jahren abstottern musste.

Ich kaufte den Rechner in einem dieser kleinen Geschäfte um die Ecke, so einer Art Tante-Emma-Laden für Computer, nur dass es in diesem Fall eher ein Onkel-Egon-Laden war. Die Marke war mir egal, und deshalb entschied ich mich für einen No-Name-Rechner in einem nichts sagenden hellen Gehäuse. Der Typ zeigte einem eine Preisliste, ein Smorgasbord, aus dem man sich die gewünschte CPU, den Speicherausbau, die Plattengröße usw. herausuchen konnte. Mir kam es auf Leistung an. Ich wollte 4 Megabyte RAM statt 2 Megabyte. Ich wollte 33 Megahertz. Klar, ich hätte mich mit 16 Megahertz zufrieden geben können, aber nein, ich wollte das Beste vom Besten.

Man sagte dem Verkäufer, was man haben wollte, und er baute die Kiste zusammen. Heute, in der Ära von Internet und UPS-Versand, klingt das kurios. Drei Tage später sollte alles fertig sein, aber diese drei Tage kamen mir vor wie eine Woche. Am 5. Januar spannte ich meinen Vater ein, mir beim Heimtransport zu helfen.

Die Kiste war nicht nur narren-, sondern auch gesichtslos. Sie war einfach ein hellgrauer Kasten. *Diesen* Computer kaufte ich nicht

wegen seines coolen Aussehens. Er war eine farblose Maschine mit einem 14-Zoll-Monitor, der billigste, einigermaßen handfeste Rechner, den ich finden konnte. Mit „handfest“ meine ich einen leistungsstarken Computer, wie ihn nicht viele Leute besaßen. Ich will damit nicht sagen, er hätte die unattraktive Funktionalität eines VolvoKombis besessen. Aber es war mir einfach wichtig, etwas Zuverlässiges zu haben, in das ich die Upgrades, die ich unweigerlich brauchen würde, leicht einbauen konnte.

Der Computer wurde mit einer abgespeckten DOS-Version ausgeliefert. Ich bestellte die Unix-Variante Minix, die ich darauf installieren wollte, und wartete dann über einen Monat darauf, bis das Betriebssystem seinen Weg nach Finnland fand. Oh ja, man konnte zwar das Buch über Minix in einem Computerladen kaufen, aber da das Betriebssystem selbst so wenig gefragt war, musste man es über den Buchhandel bestellen. Es kostete 169 Dollar plus Steuern plus Bankspesen plus alles-mögliche-andere. Ich fand (und finde!) den Preis unverschämt hoch. Der verlorene Monat kam mir vor wie sechs Jahre. Ich war noch frustrierter als in all den Monaten, in denen ich darauf wartete, meinen PC kaufen zu können.

Und das mitten im tiefsten Winter. Jedes Mal, wenn du dich aus der Wohnung hinaus ins feindliche Leben wagtest, bestand die Gefahr, von irgendeiner alten Dame in den Schnee gefegt zu werden, die besser zu Hause geblieben wäre, um Kohlsuppe für ihre Familie zu kochen oder Eishockey im Fernsehen anzuschauen und Pullover zu stricken, statt die Mannerheimintie entlang zu wanken. Ich habe den Monat im Wesentlichen damit zugebracht, *Prinz von Persien* auf meinem neuen Computer zu spielen. In der übrigen Zeit las ich Bücher, um mich in meine Neuerwerbung einzuarbeiten.

Endlich, an einem Freitagnachmittag, kam Minix an, und ich installierte es noch am selben Abend. Dazu musste man den Computer mit sechzehn Floppydisks füttern. Das ganze Wochenende über war ich damit beschäftigt, mich an das neue System zu gewöhnen. Ich lernte die Vorzüge des Betriebssystems kennen. Und, wichtiger noch, seine Nachteile. Ich versuchte, die Schwachpunkte auszugleichen, indem ich Programme vom Uni-Computer herunterlud, an die ich mich gewöhnt hatte. Insgesamt brauchte ich einen Monat oder noch länger, um mir das System zu Eigen zu machen.

Andrew Tanenbaum, der Amsterdamer Professor, der Minix geschrieben hatte, wollte das Betriebssystem als Lernhilfe einsetzen. Deshalb wurden Teile seiner Funktionalität absichtlich verstümmelt, und die Folgen waren verheerend. Es gab Patches für Minix - Verbesserungen -, unter anderem den eines australischen Hackers namens Bruce Evans, dem Gott von Minix 386. Dank seiner Verbesserungen ließ sich Minix leichter auf einem 386er einsetzen. Schon bevor ich den Computer kaufte, verfolgte ich die Minix-Newsgroups im Netz und wusste deshalb von Anfang an, dass ich diese verbesserte Version nutzen wollte. Aber wegen der Lizenzsituation musste man die Original-Minix-Version kaufen und dann in mühseliger Kleinarbeit Evans

Ausbesserungen bootstrappen. Das war ein ziemlicher Aufwand.

Minix hatte eine Reihe von Eigenschaften, die mich enttäuschten. Der größte Frust aber war die Terminal-Emulation. Sie war wichtig, weil ich damit den Uni-Computer auf meinem Heimcomputer imitieren wollte. Ich setzte die Terminal-Emulation ein, um mich in den Unicomputer einzuloggen und auf dem mächtigen Unix-Rechner zu arbeiten oder einfach online zu gehen.

Deshalb entschloss ich mich, mir selbst ein Terminal-Emulationsprogramm zu schreiben. Ich wollte das Projekt nicht unter Minix durchführen, sondern auf der nackten Hardwareebene. Das Terminal-Emulationsprojekt würde mir darüber hinaus eine großartige Gelegenheit bieten, mich mit der Arbeitsweise der Hardware des 386er vertraut zu machen. Wie gesagt, es war Winter in Helsinki. Ich hatte einen handfesten Computer. Ich sah das Projekt einfach als Möglichkeit, die Leistungsfähigkeit der Kiste kennen zu lernen und meinen Spaß zu haben.

Weil ich auf der nackten Hardware programmierte, musste ich vom BIOS ausgehen, dem ROM-Code, in den der Computer bootet. Es liest die Floppydisk oder die Festplatte. Bei mir war es eine Floppydisk. Es liest den ersten Sektor der Floppy und verzweigt dorthin. Weil ich noch nie vorher auf einem PC gearbeitet hatte, musste ich diesen Ablauf erst kennen lernen. Der 386er startet im so genannten *Real Mode*. Aber um die CPU voll auszunutzen, und in den 32-Bit-Modus zu gelangen, musst du in den *Protected Mode* kommen. Dazu ist ein komplizierter Setup erforderlich.

Um ein Terminal-Emulationsprogramm zu schreiben, muss man wissen, wie die CPU arbeitet. Tatsächlich hatte ich mich auch deshalb für

die Assemblersprache entschieden, um mehr über die CPU zu lernen. Ansonsten muss man noch wissen, wie man auf den Bildschirm ausgibt, wie man Tastatureingaben liest und wie man lesend und schreibend auf das Modem zugreift. Hoffentlich vergraulen hier nicht endgültig die Nicht-Geeks, die bisher tapfer durchgehalten haben, statt gleich auf Seite 129 weiterzulesen.

Ich wollte zwei voneinander unabhängige Threads verwenden. Der eine Thread oder Kontrollfaden sollte vom Modem lesen und das Gelesene auf dem Bildschirm ausgeben. Der andere Thread sollte von der Tastatur lesen und auf das Modem schreiben. Und es sollte zwei Pipes geben, für jede der beiden Richtungen eine. Dieses Konzept heißt Taskwechsel, und ein 386er hatte die Hardware, es zu unterstützen. Ich fand die Idee cool.

Mein erstes Testprogramm war so geschrieben, dass es einen Thread nutzte, um den Buchstaben A auf den Bildschirm zu schreiben. Der andere Thread schrieb den Buchstaben B. Ich weiß, das klingt nicht eben beeindruckend. Das Ganze war so programmiert, dass es mehrmals pro Sekunde passierte. Mit dem Timer-Interrupt sorgte ich dafür, dass der Bildschirm sich mit AAAAAAA füllte. Urplötzlich schaltete er dann aufBBBBBB um. Diese Übung hat nicht den geringsten praktischen Nutzen. Aber mir bot sie eine gute Möglichkeit, mir die Funktionsfähigkeit meines Taskwechsels zu beweisen. Ich brauchte dafür etwa einen Monat, weil ich nicht erst in das ganze Drum und Dran einarbeiten musste.

Schließlich war ich dann so weit, die beiden Threads, AAAAAAA undBBBBBB, dahin zu bringen, dass einer vom Modem las und auf den Bildschirm schrieb und der andere von der Tastatur las und auf das Modem schrieb. Ich hatte mein eigenes Terminal-Emulationsprogramm entwickelt.

Um Nachrichten zu lesen, führte ich mein Programm aus, legte meine Floppydisk ein, bootete die Kiste neu und konnte Nachrichten vom Uni-Computer lesen. Um Änderungen vorzunehmen und das Terminal-Emulationspaket zu verbessern, bootete ich Minix und nutzte es als Programmierumgebung.

Ich war sehr stolz darauf.

Sara kannte mein persönliches Meisterstück. Ich führte es ihr vor, sie schaute sich die Bildschirme mit den AAAAAAAs undBBBBBBs ungefähr fünf Sekunden lang an, sagte „Gut“ und ging aus dem Zim-

mer, vollkommen unbeeindruckt. Mir wurde klar, dass die Sache nach nichts aussah. Es ist völlig unmöglich, einem anderen Menschen zu erklären, dass etwas, das nach nichts aussieht, im Hintergrund eine Menge leistet. Es ist ungefähr so beeindruckend, wie wenn du jemandem eine frisch geteerte Straße zeigst. Wahrscheinlich war Lars, der andere schwedisch sprechende Informatikstudent, der im gleichen Jahr wie ich mit dem Studium begonnen hatte, der einzige Mensch, der meine Großtat zu würdigen wusste.

Es war März oder vielleicht April, und wenn der Schnee in der Petersgatan sich in Matsch verwandelt haben sollte, habe ich es weder gemerkt, noch hat es mich sonderlich interessiert. Ich verbrachte meine Zeit hauptsächlich im Bademantel, über meinen unattraktiven neuen Computer gebeugt, hinter dichtgewebten schwarzen Vorhängen, die mich gegen die Sonne und vor allem gegen die Außenwelt abschirmten. Ich kratzte die monatlichen Raten für meinen PC zusammen, der in drei Jahren abgezahlt werden sollte. Damals wusste ich noch nicht, dass ich die Raten nur noch ein Jahr lang würde aufbringen müssen. Da nämlich würde ich Linux geschrieben und sehr viel mehr Leute als Sara und Lars würden es gesehen haben. Und Peter

Anvin, der heute wie ich bei Transmeta arbeitet, würde eine Sammlung im Internet gestartet haben, um das Geld für meinen Computer aufzutreiben.

Das ergab sich einfach so. Jeder wusste, dass ich mit Linux kein Geld verdiente. Und plötzlich hieß es: Lass uns sammeln, um Linus' Computer abzubezahlen.

Es war überwältigend.

Ich hatte absolut kein Geld. Es war mir immer wichtig gewesen, nicht um Geld zu bitten oder zu betteln, aber dass ich es einfach so bekam, war ...

Es schnürt mir die Kehle zu.

So wurde Linux geboren. Mit meinen Testprogrammen, die sich zu einem Terminal-Emulationspaket entwickelten.

Die Zeitschrift Red Herring schickt mich nach Finnland, um eine Reportage über Oulu zu schreiben, das aufstrebende Hightech-Zentrum, das trotz seiner unwirtlichen Lage gerade mal ein paar Stunden Fahrt vom Polarkreis entfernt 141 Start-up-Firmen beherbergt. Für mich ist das eine willkommene Gelegenheit, mich mit Linus' Eltern und seiner Schwester Sara in Helsinki in Verbindung zu setzen.

Nils, den alle Nicke nennen, treffe ich in der Lobby des Sokos Hotel Vaakuna gegenüber dem Helsinkier Bahnhof. Er sieht gut aus, trägt eine dicke Brille und hat einen Lenin-Bart. Er ist erst vor kurzem von einem zehnjährigen Auslandseinsatz in Moskau zurückgekehrt, wo er für die finnische Nachrichtenagentur tätig war, schreibt jetzt ein Buch über Russland und denkt darüber nach, ob er eine Stelle in Washington, einer Stadt, die er ziemlich uninteressant findet, übernehmen soll oder nicht. Vor ein paar Monaten gewann er einen angesehenen nationalen Journalismuspreis, eine Auszeichnung, die ihn, wie seine ExFrau Anna später kommentieren wird, „erheblich milder stimmte“.

Am frühen Abend fahren wir in seinem Volvo V40 zusammen die schneeverkrusteten Wahrzeichen von Linus' Kindheit ab: Er zeigt mir das solide Gebäude, in dem sowohl Vater als auch Sohn ihre Grundschulzeit absolvierten, fährt an der Wohnung der Großeltern vorbei, wo Linus seine ersten drei Lebensmonate verbrachte, und zeigt mir dann das Gebäude mit dem Parkblick, wo die Familie in den folgenden sieben Jahren wohnte. Nicke lebte ein Jahr davon in Moskau, wo er lernte, Kommunist zu werden. Linus war damals fünf Jahre alt. Als Nächstes weist er mich auf das blassgelbe Wohnhaus hin, in das Linus und seine Schwester nach der Scheidung zogen - im Erdgeschoß hat eine Videothek für Erwachsene das Elektrogeschäft aus Linus' Jugend abgelöst -, und schließlich fahren wir an dem gediegeendsten dieser Bauwerke vorbei, einem fünfstöckigen Wohnhaus, in dem früher die Großeltern mütterlicherseits wohnten, dem Geburtsort von Linus. Anna lebt immer noch dort. Ein bisschen wirkt es wie die Upper East Side von Manhattan an einem der letzten Tage im Dezember.

Nicke ist witzig, intelligent, selbstironisch und hat eine Menge Gesten mit seinem Sohn gemeinsam, zum Beispiel die Art, wie er beim Reden das Kinn in der Hand wiegt. Sie haben das gleiche Grinsen. Im Gegensatz zu seinem Sohn ist er aktiver Sportler - ein sozialistischer Jock - und spielt in einer Basketballmannschaft, läuft täglich acht Kilometer und hat sich angewöhnt, jeden Morgen lange Strecken in einem eisigen See zu schwimmen. Mit 22 bewegt er sich mit dem athletischen Selbstvertrauen eines Mittdreißigers. Und noch etwas teilt er nicht mit Linus: Nicke scheint ein kompliziertes Liebesleben hinter sich zu haben.

Wir essen in einem gut besuchten Restaurant im Zentrum von Helsinki zu Abend, wo Nicke davon erzählt, wie schwer es für Linus war, als Sohn eines hyperaktiven Kommunisten aufzuwachsen - Nicke hielt häufig Reden und hatte einmal sogar ein kleines öffentliches Amt inne. Er erzählt, dass Linus oft wegen der radikalen Politik seines Vaters gehänselt wurde, und dass manche Eltern sich sogar weigerten, ihn mit ihren Kindern spielen zu lassen. Aus diesem Grund, erklärt Nicke, war sein Sohn stets bemüht, sich von der linken Rhetorik, der Geräuschkulisse seiner Kindheit, zu distanzieren. „Er ließ keine Diskussionen darüber zu. Er ging einfach aus dem Zimmer“, erzählt er. „Oder er vertrat aus Prinzip einen gegensätzlichen Standpunkt. Ich weiß, dass Linus in der Schule damit zu kämpfen hatte, dass er den falschen Vater hatte. Seine Botschaft an mich war: ‚Erspar mir diese peinliche Situation.‘

Nicke fährt mit mir in seine Wohnung und lädt mich zu einem Bier in der Küche ein. Seine Wohnung liegt mitten im kommerziellen Zentrum von Helsinki, in einem Arbeiter-Wohnblock aus den 1920er-Jahren. Wir steigen die Treppe zu seiner Wohnung hinauf und ziehen im Flur unsere Schuhe aus. Die Wohnung erinnert mit ihren Lampen aus Peddigrohr, dem Wandschmuck aus der Dritten Welt und den Zimmerpflanzen an die Gegenkultur der 1960er-Jahre. „Eltern sollten nicht denken, sie würden ihre Kinder formen“, sagt er und greift nach seinem Mobiltelefon, um seine Lebensgefährtin anzurufen. Er erzählt, Linus habe erst jetzt angefangen, die Bücher über Geschichte zu lesen, die er ihm seit Jahren aufdränge, und sich vermutlich nie die Mühe gemacht, die Gedichte seines eigenen Großvaters zu lesen.

Ich frage Nicke, ob er je ein Interesse an Programmierung gezeigt hat, ob er Linus je gebeten hat, ihm die Grundlagen beizubringen. Nein, sagt er. Väter und Söhne sind einzigartige Individuen. Sich auf die glei-

che Leidenschaft zu stürzen wie Linus käme einem „Eindringen in seine Seele“ gleich. _ Er scheint sich in seiner Rolle als Vater eines berühmten Menschen wohl zu fühlen. Angesichts der Verleihung des nationalen Journalismuspreises erschien kürzlich ein Zeitungsporträt über ihn, dem zufolge er gesagt haben soll, schon als er Linus vom Spielplatz abholte, hätten die anderen Kinder auf ihn gezeigt und gerufen: „Schaut, dort kommt der Vater von Linus.“

.. ..

Sara Torvalds ist mit dem Zug aus der kleinen Stadt westlich von Helsinki gekommen, in der sie wohnt. Dort sind die Straßenschilder zuerst schwedisch und dann finnisch beschriftet, dort kann sie sich eine Wohnung mit einer Badewanne mit Krallen Füßen und Sauna leisten und dort hört man zu ihrer großen Freude auf den Straßen schwedisch und nicht finnisch. Sie erklärt mir, dass sie einer Minderheit innerhalb einer Minderheit angehört: Als junge Erwachsene konvertierte sie zum katholischen Glauben und traf damit eine Entscheidung, die sie in die Gruppe der 10 Prozent der finnischen Bürger verwies, die nicht lutheranisch sind, was ihren agnostischen Vater veranlasste, sie für ein paar Wochen zu enterben.

Heute ist sie nach Helsinki gekommen, um Jugendlichen im Rahmen eines von der Regierung geförderten Programms katholischen Religionsunterricht zu erteilen. Sie ist frisch und sympathisch, und strahlt trotz ihrer 29 Jahre die Unverdorbenheit einer ernsthaften und viel beschäftigten Gymnasiastin aus. Mit ihrer hellen Haut und ihrem runden Gesicht besitzt sie eine vage Ähnlichkeit mit ihrem älteren Bruder. Aber es ist offensichtlich, dass sie von Natur aus viel kontaktfreudiger ist als er. Immer wieder tippt sie auf der Tastatur ihres Handys herum, um SMSs an Freunde zu schicken, die sie später treffen wird, und ebenso häufig schaut sie nach, ob Antworten gekommen sind. Sie hat ein erfolgreiches Übersetzungsbüro.

Es ist Mittag, und Sara nimmt mich zu ihrer Mutter zum Essen mit. Auf dem Weg dorthin legen wir diverse Zwischenstopps an den Schauplätzen ihrer Kindheit ein: dem Katzenpark, der Grundschule. „Meine Eltern waren Kommunisten mit Parteibuch, und so wurden wir erzogen. Zu glauben, dass die Sowjetunion eine gute Sache ist. Sie sind mit uns nach Moskau gefahren“, erzählt sie. „Am meisten hat mich das

riesige Spielzeuggeschäft beeindruckt, das es dort gab. Es war größer als alles, was wir aus Helsinki kannten.“ Ihre Eltern ließen sich scheiden, als sie sechs war. „Ich erinnere mich daran, wie sie uns sagten, dass Papa für immer ausziehen würde. Ich dachte, das ist gut. Jetzt hören sie endlich auf, sich zu streiten. Er war ohnehin immer wieder über lange Phasen hinweg in Moskau gewesen, so dass wir daran gewöhnt waren, dass er nicht da war“, sagt sie. Mit zehn entschied Sara sich dafür, zu ihrem Vater zu ziehen, der mittlerweile in der Nachbarstadt Espoo lebte, statt weiter bei ihrer Mutter und Linus zu wohnen. „Es war nicht so, dass ich nicht bei Mama leben wollte. Es war nur so, dass ich nicht mit Linus leben wollte. Auf diese Weise konnten wir uns nur am Wochenende streiten. Wir stritten uns ständig. Erst als wir älter wurden, legte sich das allmählich.“

Wir kommen in der Wohnung ihrer Mutter im ersten Stock an, wo Anna Torvalds uns begeistert in Empfang nimmt. Mikke ist ihr Kosenamen. Sie lässt nicht zu, dass ich die Schuhe ausziehe, wie es in Finnland üblich ist. „Jetzt kommen Sie schon rein. Die Wohnung ist ohnehin schon staubig. Schlimmer kann es nicht werden.“ Sie ist klein, dunkelhaarig und überaus schlagfertig. Sekunden nach unserer Ankunft klingelt das Telefon. Eine Maklerin soll mir die leer stehende Wohnung neben Mikkes Wohnung zeigen, so dass ich sie ihrem Sohn in den USA beschreiben und ihm persönlich die entsprechenden Unterlagen überbringen kann, für den Fall, dass er sie als eine Art Zweitwohnung in Helsinki kaufen möchte. Wir betreten die großzügige Wohnung, und die Maklerin, die auf gespenstische Weise der Figur ähnlich sieht, die Annette Bening in American Beauty verkörpert hat, fordert uns auf, vor der Besichtigung der Räume blaue kleine Strümpfe über unsere Schuhe zu ziehen. Es dauert nicht lange, bis die Maklerin mit irritierend munterer Stimme etwas in der Art sagt wie: „Dann kommen wir zu diesem Zimmer hier. Es eignet sich perfekt für Antiquitäten, die Sie gegen zu starke Sonneneinstrahlung schützen möchten.“ Mikke wirft mir einen verschwörerischen Blick zu und antwortet spöttisch: „Oh, was für eine reizende Art, uns wissen zu lassen, dass dieses Zimmer kein Licht bekommt.“

Später sitzen wir dann an dem rechteckigen Tisch in Mikkes Wohnung, auf dem ein farbenfrohes Tischtuch liegt, und Mikke gießt sich Kaffee in einen übergroßen Becher ein. Ihre Wohnung ist wie die ihres Ex-Mannes mit Büchern und ethnischer Kunst angefüllt. An den Fens-

tern hängen schwarz-weiße Vorhänge mit Marimekko-Muster. Die Wohnung bestand ursprünglich aus drei Schlafzimmern und einer Küche. Als ihre Kinder auszogen, übersiedelte Mikke in das große Schlafzimmer, das Sara gehört hatte. Dann riss sie die Wände zu Linus' Zimmer und ihrem ehemaligen Schlafzimmer ein, um auf diese Weise ein riesiges Wohnzimmer mit offener Küche zu erhalten. Sie zeigt auf eine leere Stelle und sagt: „Dort stand sein Computer. Wahrscheinlich sollte ich dort eine Plakette oder so anbringen. Was halten Sie davon?“ Sie raucht unablässig. Sie plaudert gewandt, und ihr Englisch ist so gut, dass ihr Sätze wie „Er ist kein 08/1 S-Typ, wie man ihn an jeder Straßenecke trifft“ mühelos über die Lippen gehen. An der Wand in ihrem Schlafzimmer hängt eine riesige Sowjet-Flagge. Jouko Vierumaki hatte sie während eines internationalen Wettbewerbs im Skispringen erstanden und Linus mitgebracht. Linus hatte die Fahne jahrelang in einer Schublade aufbewahrt, bis Mikke sie über ihr Bett hängte.

Mizke zieht ein Album heraus, in das die wenigen Familienfotos eingeklebt sind. Linus im Alter von zwei oder drei Jahren, nackt am Strand. Linus im gleichen Alter, wie er verloren vor einem berühmten Schloss in der Nähe von Helsinki steht. Linus als dünner, ungelinker Teenie. Mikke beim 60. Geburtstag ihres Vaters, dem Statistikprofessor. Sie zeigt mir ihre Schwester und ihren Bruder, die beide älter sind als sie. „Sie ist Psychiaterin in New York. Er ist Kernphysiker. Und ich, ich bin das schwarze Schaf. Stimmts? Aber ich hatte das erste Enkelkind“, berichtet sie und zündet sich eine Gauloise an.

Wir essen in einem Restaurant zu Mittag, das nach Wilt Chamberlain benannt ist. Sara checkt ihr Handy, während Mikke sich mehrere Espressos bestellt. Mikke erinnert sich daran, wie sie und Nicke darüber stritten, ob sie Linus zwingen sollten oder nicht, auf seinen Schnuller zu verzichten: Sie schrieben einander kurze Mitteilungen und ließen sie auf der Küchentheke liegen. Wir reden über Linus' schlechtes Gedächtnis und seine Unfähigkeit, sich an Gesichter zu erinnern. „Wenn du dir mit ihm einen Film ansiehst, und der Held sein rotes Hemd gegen ein gelbes auswechselt, fragt Linus ‚Und wer ist dieser Typ?‘, sagt Sara. Wir reden über einen Camping-Fahrrad-Urlaub der Familie in Schweden. Die Übernachtung auf dem Fährschiff. Wie gleich am ersten Tag Saras Fahrrad gestohlen wurde. Wie das Ferienbudget für ein neues Rad draufging. Wie sie das Zelt auf einer Klippe

aufbauten. Wie Linus den ganzen Tag lesend im Zelt verbrachte, während Mutter und Tochter schwammen und fischten. Und wie sie dann nach einem starken Sturm begriffen, dass das Zelt nur aus einem einzigen Grund nicht in die Ostsee geweht worden war: wegen Linus, der den extremen Wetterumschwung seelenruhig darin verschlafen hatte.

Mikke lacht, als sie die Jahre Revue passieren lässt, in denen Linus sich in seinem Zimmer verkroch und über seinem Computer brütete. „Nicke lag mir dauernd in den Ohren ‚Schmeiß ihn raus, sorg dafür, dass er sich einen Job sucht‘, aber Linus störte mich nicht. Er war anspruchslos. Und was immer er da mit seinem Computer machte, war seine Sache, sein Ding, und sein gutes Recht obendrein. Ich hatte keine Ahnung, worum es dabei ging.“ Heute ist sie über die Aktivitäten ihres Sohnes so gut informiert wie jeder andere. Mikke und die anderen Familienmitglieder werden ständig mit Anfragen der Medien bombardiert. Die Anfragen werden an Linus weitergeleitet, der normalerweise seine Mutter, seinen Vater oder seine Schwester auffordert, nach eigenem Ermessen zu antworten. Trotzdem schicken sie die fertige Antwort meistens erst an Linus, damit er sie absegnen kann, bevor sie ein Reporter bekommt.

Vor einigen Monaten, als ich Mikke in einer E-Mail um ihre Erinnerungen an Linus' Kindheit bat, bekam ich eine ausführliche, liebevoll formulierte Antwort. Ihr Essay war mit dem Titel überschrieben:

„Wie ein winzigkleiner Nerd zu Linus heranwuchs“. Darin erzählte sie, wie sie merkte, dass ihr kleiner Sohn die gleichen Anzeichen wissenschaftlicher Entschlossenheit zeigte, die sie von ihrem Vater und ihrem älteren Bruder kannte:

„Wenn du einen Menschen siehst, der glasige Augen bekommt, sobald ein Problem auftritt oder ihn beschäftigt, der nicht hört, was andere sagen, der nicht einmal die einfachste Frage beantwortet, der völlig aufgeht in der Aufgabe, die ihn gefangen nimmt, der während der Suche nach einer Lösung bereitwillig auf Essen und Schlaf verzichtet und der nicht aufgibt. Nie. Ungeachtet aller alltäglichen Unterbrechungen kommt er - oder natürlich auch sie - immer wieder hartnäckig darauf zurück. Dann weißt du es.“

Sie schilderte die Geschwisterrivalität zwischen Linus und Sara und die unvereinbaren Gegensätze zwischen ihnen. (Sara: „Ich MAG den Geschmack von Pilzen/Leber/Was-auch-immer nicht.“ Linus: „TUST du DOCH!“) Und den widerwilligen Respekt. „Linus brachte einmal,

als er noch ziemlich klein war, seine Achtung vor seiner Schwester sehr klar zum Ausdruck. Er mag damals fünf oder sieben oder so gewesen sein, als er mir sehr ernst sagte: „Weißt du was? Ich denke keine neuen Gedanken. Ich denke Gedanken, die andere Leute gedacht haben, und dann ordne ich sie neu an. Aber Sara, sie denkt Gedanken, die es nie vorher gegeben hat.“

Vielleicht sehen Sie an diesen Erinnerungen, dass ich noch immer nicht glaube, Linus habe eine außergewöhnliche Begabung für Computer - wären es nicht die Computer, wäre es etwas anderes. In einem anderen Zeitalter hätte er sich auf eine andere Herausforderung gestürzt, und ich denke, eines Tages wird er das auch noch tun. (Ich will damit sagen, dass ich hoffe, dass er nicht sein Leben lang an Linux und dessen Pflege hängen bleiben wird). Meiner Meinung nach ist er nämlich nicht durch „Computer“ motiviert, und ganz gewiss nicht durch Ruhm und Reichtum, sondern durch echte Neugierde und den Wunsch, auftretende Schwierigkeiten zu bewältigen, und zwar auf die richtige Weise, weil es ihm nur so sinnvoll erscheint. Vorher lässt er nicht locker.

Ich glaube, die Frage, wie Linus als Sohn war, habe ich schon beantwortet - leicht zu erziehen, ja. Er brauchte nur eine Herausforderung, das Übrige besorgte er alleine. Als er dann anfing, sich auf Computer zu konzentrieren, wurde es noch einfacher. Sara und ich sagten immer, um Linus glücklich zu machen, brauchst du ihn nur in eine Kammer mit einem guten Computer darin zu setzen und ihm ein paar gekochte Nudeln zu essen zu geben.

Um eines allerdings machte ich mir Sorgen, als er größer wurde: Wie um alles in der Welt sollte er auf diese Weise jemals nette Mädchen kennen lernen? Mir blieb nichts anderes übrig, als mich wieder einmal auf die bewährte und erprobte elterliche Maßnahme zu verlassen, die Daumen zu drücken. Und, halleluja: Es hat gewirkt. Er traf Tove, als er Kurse an der Uni gab, und sie ließ ihn ein paar Tage lang sowohl seine Katze als auch seinen Computer vergessen. Man sah sofort, dass letztlich doch die Natur, wie es ihre Gewohnheit ist, den Sieg davon getragen hatte.

Ich hoffe nur, dass die Geister des Erfolgs ihm nicht zu Kopf steigen. (Der Ruhm scheint ihn nicht verändert zu haben, aber er ist zugänglicher geworden, und redet jetzt mit Leuten, die ihn anspre-

chen. Es scheint ihm sogar schwer zu fallen, nein zu sagen. Aber ich vermute, das hat mehr damit zu tun, dass er geheiratet hat und Vater geworden ist, als mit all dem Medienrummel.) "

Es ist offensichtlich, dass sowohl Mutter als auch Tochter dem Medienrummel gelassen begegnen. Einen Tag vor unserem Treffen hatte Transmeta öffentlich bekanntgegeben, was es da eigentlich auskochte, und während unseres Mittagessens fragt Mikke Sara: „Stand heute etwas über Du-weißt-schon-wer und Du-weiß-schon-was in der Zeitung?“

Am gleichen Abend lässt Mikke auf dem Weg zur Arbeit das Taxi vor meinem Hotel warten, um mir einen Kinderstuhl aus Kiefernholz zu übergeben, den ich Patricia persönlich überbringen soll. Das Stühlchen und einen Grundriss der frei gewordenen Wohnung.

Meine erste Erinnerung an Linus' Heldenstück
ich glaube, es war Anfang 1992. Ich besuchte Linus mal wieder in seiner
völlig unaufgeräumten Wohnung - mit dem Fahrrad und ohne
besonderen Anlass. Während ich, wie gewöhnlich, MTV guckte,
erkundigte ich mich bei Linus nach seiner Betriebssystementwicklung.
Normalerweise gab er mir eine nichts sagende Antwort. Dieses Mal
führte er mich an seinen Computer (aus der unaufgeräumten Küche der
Torvaldsens in sein völlig chaotisches Zimmer).

Linus gab seinen Benutzernamen und sein Passwort ein und kam zu
einem Eingabeprompt. Er zeigte mir einige Grundfunktionen des
Kommandointerpreters, die aber nicht weiter aufregend waren. Nach
einer Weile wandte er sich mir mit einem LinusGrinsen auf dem
Gesicht zu und fragte: „Es sieht wie DOS aus, oder?“

ich war beeindruckt und nickte. Aber ich war nicht überwältigt, weil es
zu sehr nach DOS aussah - und eigentlich nichts Neues bot. Ich hätte
wissen sollen, dass Linus nie ohne guten Grund auf diese Weise grinst.
Linus wandte sich wieder seinem Computer zu und drückte eine
Funktionstasten-Kombination - ein anderer Login-Bildschirm wurde
angezeigt. Ein neuer Login und ein neuer Eingabeprompt. Linus zeigte
mir vier verschiedene Eingabeprompts und erklärte mir, später würden
vier unterschiedliche Anwender darauf zugreifen können.
In diesem Moment wusste ich, dass Linus etwas Phantastisches
geschaffen hatte. Das ist kein Problem für mich - am Billard-Tisch bin
immer noch ich die Nummer eins.

Jouko, Avuton' Vierumaki

„Für mich bedeutete es hauptsächlich, dass das Telefon ständig belegt
war und uns niemand anrufen konnte ... Irgendwann begannen
Postkarten aus allen möglichen Ecken der Welt bei uns einzutreffen.
Damals muss mir klar geworden sein, dass es in der realen Welt
Menschen gab, die das, was er da geschaffen hatte, tatsächlich
benutzten.“

Sara Torvalds

Die Schönheit des Programmierens

Ich weiß nicht, wie ich erklären soll, was mich am Programmieren
so fasziniert, aber ich werde es versuchen. Für jemanden, der
programmiert, ist es das Interessanteste auf der Welt. Es ist ein
Spiel, das dich viel mehr fesselt als Schach, bei dem du deine
eigenen Regeln aufstellen kannst, und bei dem am Ende das
herauskommt, was du daraus machst.

Und trotzdem sieht es nach außen hin wie die langweiligste Sache
der Welt aus.

Ein Aspekt des anfänglichen Reizes ist schnell erklärt: Er ergibt sich
ganz einfach **aus der Tatsache, dass der Computer das tut, was
du ihm sagst**. Unbeirrbar. Für immer. Ohne ein Wort der Klage.

Und schon das ist interessant.

Aber blinder Gehorsam allein, so faszinierend er zunächst auch sein
mag, ist auf Dauer nicht besonders reizvoll. Deshalb wird einem
dieser Aspekt auch ziemlich schnell ziemlich langweilig. Das
wirklich Fesselnde am Programmieren ist die Tatsache, dass du den
Computer dazu bringen kannst, zu tun, was du willst, aber du musst
herausfinden, *wie*.

Ich persönlich bin überzeugt davon, dass Informatik und Physik viel
gemeinsam haben. Beide beschäftigen sich damit, was die Welt im
Innersten zusammenhält. Der Unterschied liegt natürlich darin, dass
du in der Physik herausfinden sollst, nach **welchen Gesetzen die
Welt funktioniert, während** du in der Informatik die Welt
erschaffst. Innerhalb der Grenzen deines Computers bist du der
Schöpfer. Irgendwann bist du so weit, dass du das Geschehen
komplett unter Kontrolle hast.

Wenn du gut genug bist, kannst du Gott sein. Jedenfalls in einem bescheidenen Rahmen..

Wahrscheinlich bin ich mit dieser Aussage gerade ungefähr der Hälfte der Erdbevölkerung zu nahe getreten.

Das ändert nichts an ihrem Wahrheitsgehalt. **Du kannst dir deine eigene Welt erschaffen, und die einzigen Faktoren, die dich in deinen Möglichkeiten einschränken, sind die Fähigkeiten der Kiste - und, mehr als je zuvor, dein eigenes Können.**

Stellen Sie sich ein Baumhaus vor. **Sie können ein Baumhaus bauen, das funktional und stabil ist und eine Klapptür hat.** Aber jeder erkennt den Unterschied zwischen einem Baumhaus, das einfach solide gebaut ist, und einem **Baumhaus, das schön ist und die Besonderheit des Baumes kreativ zu nutzen versteht.** Es ist eine Frage der geschickten Kombination von Kunst und Technik. Das ist einer der Gründe, der das Programmieren so fesselnd und lohnenswert erscheinen lässt. Die Funktionalität ist oft zweitrangig; viel wichtiger ist es, etwas Interessantes, Schönes oder Schockierendes zuwege zu bringen.

Programmieren ist eine Übung in Kreativität.

Ich bin überhaupt erst zum Programmieren gekommen, weil ich herausfinden wollte, wie der Computer arbeitet. Zu meiner größten Freude entdeckte ich, dass es sich mit Computern verhält wie mit der Mathematik: Du kannst dir deine eigene Welt mit ihren eigenen Regeln aufbauen. In der Physik wirst du von vorhandenen Regeln eingeschränkt. Aber in der Mathematik und beim Programmieren ist alles möglich, solange es in sich konsistent ist. Die Mathematik braucht sich keiner Beschränkung durch eine äußere Logik zu unterwerfen, aber sie muss in sich selbst und aus sich selbst heraus logisch sein. Wie jeder Mathematiker weiß, kann man eine Menge mathematischer Gleichungen aufstellen, bei denen drei plus drei gleich zwei ist. Man kann alles tun, was man will, aber je mehr Komplexität man hinzufügt, desto sorgfältiger muss man darauf achten, Inkonsistenzen mit der geschaffenen Welt zu vermeiden. Wenn das Geschaffene schön sein soll, darf es keine Fehler enthalten. Das ist das Wesen des Programmierens.

Die Leute sind von Computern unter anderem deshalb so begeistert, weil sie mit ihrer Hilfe mit den geschaffenen neuen Welten experimentieren und ihre Möglichkeiten erkunden können. In der Mathematik kannst du dich in mentalen Gymnastikübungen des Möglichen ergehen. **Die meisten Leute verstehen unter Geometrie zum Beispiel eukli-**

dische Geometrie. Erst der Computer hat es ihnen ermöglicht, sich andere Geometrien bildlich vorzustellen, die nicht im Geringsten euklidisch sind. Mit Computern kannst du sehen, wie diese erfundenen Welten aussehen. Erinnern Sie sich an Mandelbrot-Mengen - die fraktalen Bilder auf der Basis von Benoit Mandelbrots Gleichungen? Sie sind visuelle Repräsentationen einer rein mathematischen Welt, die ohne die Hilfe eines Computers nicht bildlich darstellbar wäre. **Mandelbrot erfand einfach beliebige Regeln über eine Welt, die es nicht gibt und die für die Realität keine Bedeutung hat, aber dann stellte sich heraus, dass seine Regeln faszinierende Muster erzeugten. Mit Computern und Programmierung kannst du neue Welten entwickeln, und manchmal entstehen dabei wunderschöne Muster.**

Meistens tust du das aber nicht. Du schreibst einfach ein Programm, das eine bestimmte Aufgabe erledigt. **In diesem Fall erschaffst du keine neue Welt, sondern löst ein Problem innerhalb der Welt des Computers. Das Problem wird gelöst, indem man darüber nachdenkt. Dazu braucht es eine bestimmte Art von Mensch** - einen Menschen, dem es nichts ausmacht, einfach so dazusitzen, auf einen Bildschirm zu starren **und den Dingen auf den Grund zu gehen.** Dazu braucht es einen freakigen Geek wie mich.

Das Betriebssystem ist die Basis für alles andere, was in der Kiste passiert. **Und ein Betriebssystem zu entwickeln, ist die ultimative Herausforderung. Wenn du ein Betriebssystem entwickelst, entwickelst du die Welt, in der alle Programme, die auf dem Computer laufen, leben - du stellst also im Prinzip die Regeln dafür auf, was akzeptabel ist und was getan werden darf und was nicht.** Im Prinzip machst du das zwar bei jedem Programm, aber das Betriebssystem ist einfach das grundlegendste aller Programme. Es ist, als würdest du die Verfassung für ein Land schreiben, das du gerade erfindest - alle anderen Programme sind im Vergleich dazu nur gewöhnliche Gesetze.

Manchmal ergeben die Gesetze keinen Sinn. Dann setzt du alles daran, sie mit Sinn zu erfüllen. Du willst dir die Lösung ansehen können und erkennen, dass du auf die richtige Weise die richtigen Antworten gefunden hast.

Erinnern Sie sich noch an die Mitschüler, die immer die richtige Antwort wussten? Sie kamen viel schneller darauf als alle anderen, und sie mussten sich dafür noch nicht einmal anstrengen. Sie machten sich nicht die Mühe, den vorgeschriebenen Lösungsweg einzuhalten. **Statt-**

dessen betrachteten sie das Problem auf die richtige Weise. Und wenn Sie ihre Antwort hörten, wussten Sie sofort: Das ist es.

Bei Computern ist es genauso. Sie können etwas mit der Brute-Force-Methode lösen, der dummen Schufte-dich-solange-mit-dem-Problem-ab-bis-es-keines-mehr-ist-Methode. Oder Sie können den richtigen Ansatz finden, und plötzlich löst sich das Problem einfach in Luft auf. Es gibt da dieses seltsame Phänomen: Sie erkennen, dass ein Problem nur deshalb ein Problem war, weil Sie es die ganze Zeit über aus der falschen Perspektive betrachtet haben.

Das wahrscheinlich beste Beispiel dafür stammt nicht aus der Informatik, sondern aus der Mathematik. Als der große deutsche Mathematiker Carl Friedrich Gauss noch zur Schule ging, ließ angeblich ein überdrüssiger Lehrer die Schüler alle Zahlen zwischen 1 und 100 addieren - sozusagen als Beschäftigungstherapie, in der Meinung, die jungen Leute würden den ganzen Tag dafür brauchen. Der aufstrebende Mathematiker aber hatte die korrekte Antwort innerhalb von fünf Minuten gefunden: 5050. Die Lösung liegt nämlich nicht darin, dass man wirklich alle Zahlen addiert; das wäre öde und stupide. Stattdessen entdeckte er, dass man durch Addieren von 1 und 100 101 erhält. Und 2 plus 99 ergibt ebenfalls 101. Genauso wie 3 plus 98. Und SO plus 51. Es war eine Sache von Sekunden, auszurechnen, dass es SO Paare gibt, die alle 101 ergeben. Die Antwort lautet damit: 5050.

Die Geschichte mag erfunden sein, aber ihre Aussage prägt sich ein: Ein großer Mathematiker wählt nicht den umständlichen, langweiligen Weg. Stattdessen erkennt er das eigentliche Muster, das einer Frage zugrunde liegt, und nutzt es, um einen viel besseren Lösungsweg einzuschlagen. Und genau so ist es auch in der Informatik. Klar, natürlich kann man ein Programm schreiben, das die Summe berechnet. Auf einem Computer von heute ist das eine Kleinigkeit. Aber ein wirklich guter Programmierer wüsste die Antwort auch so, kraft seiner Intelligenz. Er wüsste, wie man ein schönes Programm schreibt, das das Problem auf eine neue Weise löst, die sich am Ende als die richtige erweist.

Es ist und bleibt schwer zu erklären, was so faszinierend daran sein soll, drei Tage lang mit dem Kopf gegen die Wand zu schlagen, weil man nicht weiß, wie man ein Problem auf schöne Weise lösen kann. Aber sobald du diese bessere Möglichkeit gefunden hast, ist es das schönste Gefühl der Welt.

via

Mein Terminal-Emulator nahm Gestalt an. Ich verwendete ihn regelmäßig, um mich in den Uni-Computer einzuloggen und E-Mails zu lesen oder an den Diskussionen der Minix-Newsgroup teilzunehmen. Das Problem war nur, dass ich Sachen downloaden und uploaden wollte. Zu diesem Zweck war es notwendig, Dinge auf Platte zu sichern. Und dazu brauchte mein Terminal-Emulator einen Festplattentreiber. Außerdem brauchte ich einen Dateisystemtreiber, um mir die Organisation der Platte anschauen und das heruntergeladene Zeug als Dateien speichern zu können.

An diesem Punkt hätte ich fast aufgegeben, weil ich dachte, der Aufwand würde zu groß sein und sich letztlich nicht lohnen. Andererseits gab es sonst nicht viel zu tun. Die Vorlesungen, die ich in diesem Frühjahr hörte, waren nicht besonders anspruchsvoll. Und die einzige andere Aktivität, die mich aus dem Haus führte, war das wöchentliche Spektrum-Treffen (Party) am Mittwochabend. In meiner Kontaktarmut wurde es zur einzigen Gelegenheit, bei der ich nicht programmierte oder für mein Studium arbeitete. Ohne diese Treffen (Partys) wäre ich in diesem Frühling nicht nur beinahe, sondern völlig zum Einsiedler mutiert. Spektrum lieferte sozusagen ein eingebautes soziales Leben mit, und ich glaube nicht, dass ich jemals ein Event verpasste. Die Treffen waren mir wichtig - so wichtig, dass ich manchmal in der Nacht davor wachlag und hoffte, man würde mir meine Hemmungen wegen meiner mangelnden sozialen Kompetenz, meiner Nase oder meiner nicht vorhandenen Freundin nicht anmerken. Das sind die typischen Sorgen eines Geeks.

Ich will damit sagen: Ich hatte nicht so furchtbar viele andere interessante Dinge am Laufen. Und das Festplatten-/Dateisystemtreiberprojekt versprach, interessant zu werden. Also beschloss ich, es durchzu-

ziehen. Ich schrieb einen Festplattentreiber. Und weil ich meine Dateien auf mein Minix-Dateisystem sichern wollte - und weil **das Minix-Dateisystem so gut dokumentiert war -, machte ich mein Dateisystem kompatibel zum Minix-Dateisystem.** Auf diese Weise konnte ich Dateien lesen, **die ich unter Minix erzeugt hatte, und sie auf die gleiche Platte schreiben, so dass Minix die Dateien lesen konnte,** die ich von meinem Terminal-Emulator aus erzeugte.

Das erforderte eine Menge Arbeit - einen Tagesablauf, der sich in **Programmieren, Schlafen, Programmieren, Schlafen, Programmieren, Essen (Brezeln), Programmieren, Schlafen, Programmieren, Duschen (kurz), Programmieren** erschöpfte. Zu diesem Zeitpunkt war mir klar geworden, dass das Projekt dabei war, ein Betriebssystem zu werden. Ich fing an, es mit neuen Augen zu betrachten und nicht mehr als Terminal-Emulator, sondern als Betriebssystem zu sehen. Vermutlich fand dieser Übergang während einer meiner hypnotisierenden MarathonProgrammiersitzungen statt. Am Tag oder in der Nacht? Ich weiß es nicht mehr. Im einen Moment hacke ich in meinem abgeschabten Bademantel an einem Terminal-Emulator mit Zusatzfunktionen herum. Im nächsten Moment hat er plötzlich so viele Funktionen, dass ein Betriebssystem in Arbeit daraus geworden ist.

Ich nannte es meinen „GNU-Emacs der Terminal-Emulationsprogramme“. Der GNU-Emacs war ursprünglich ein Editor gewesen, aber seine Entwickler statteten ihn mit einer Fülle von Zusatzfunktionen aus. Eigentlich hatten sie vor, einen programmierbaren Editor zu bauen, aber dann nahm die Programmierbarkeit überhand und heraus kam ein Teufelsding von Editor. Er enthält alles außer einem Spülbecken - das dafür manchmal als Icon für den Editor herhalten darf. Der GNU-Emacs gilt als ungeheure Programmieranstrengung und kann mehr, als ein Editor jemals können muss. Meinem Terminalemulator widerfuhr das Gleiche. Er wuchs über sich selbst hi-

naus.

From: torvalds@klaava.Helsinki.Fi (Linus Benedict Torvalds)
newsgroup: comp.os.minix
Subject: Gcc-1.40 und eine POSIX-Frage
Message-ID:
<1991Jul13.100050.9886@klaava.Helsinki.Fi>
Date: 3 Jul 91 10:00:50 GMT

Hallo Netzbürger, wegen eines Projekts, an dem ich arbeite (unter Minix), interessiere ich mich für die POSIX-Standarddefinition. Könnte mir bitte jemand eine (vorzugsweise) maschinenlesbare Version der neuesten POSIXRegeln nennen? Ftp-Sites wären gut.

Okay, das ist der erste öffentliche Nachweis, dass ein Geek in Finnland dabei war, die Grenzen seines Computerkönnens auszutesten. **Die POSIX-Standards sind die langatmigen Regeln zu jedem der Systemaufrufe in Unix.** Es **gibt Hunderte davon und man braucht sie, um den Computer dazu zu bringen,** seine Operationen durchzuführen, angefangen mit Read, Write, Open, Close. **POSIX ist** ein Gremium für **Unix-Standards, eine Organisation aus Vertretern von Firmen, die sich auf gemeinsame Richtlinien einigen möchten.** Standards sind wichtig, damit **Programmierer Anwendungen für das Betriebssystem schreiben können, die sich auf mehr als einer Version ausführen lassen.** Mit den Systemaufrufen - vor allem den wichtigen - würde ich eine Aufstellung der für ein Betriebssystem benötigten Funktionen besitzen. In Anlehnung daran wollte ich den Code schreiben und jede dieser Funktionen auf meine Weise realisieren. Indem ich den Standard einhielt, würden andere meinen Code nutzen können.

Ich wusste damals noch nicht, dass man diese Regeln direkt bei POSIX als **Hard-Copy kaufen konnte, aber es hätte ohnehin keine Rolle gespielt.** Selbst wenn ich sie mir hätte leisten können, dauerte es erfahrungsgemäß lange, bis Lieferungen in Finnland ankamen. Daher meine Bitte um eine Version, die ich umsonst von einer ftp-Site herunterladen konnte.

Niemand konnte mir eine Quelle für die POSIX-Standards nennen, und deshalb setze ich Plan B in Kraft. **Ich spürte die Handbücher für die Unix-Version von Sun Microsystems in der Uni auf, wo es einen Sun-Server gab. Die Handbücher enthielten eine Basisversion der Systemaufrufe, und damit konnte ich ganz gut hinkommen.** Man konnte in den Handbuchseiten nachschlagen, was ein Systemaufruf leisten sollte, und ihn dann entsprechend implementieren. Den Handbuchseiten war nicht zu entnehmen, wie das ging, nur wie das Endergebnis aussehen musste. Einige der Systemaufrufe konnte ich auch Andrew Tanenbaums Buch und ein paar anderen Büchern entnehmen. **Schließlich sandte mir jemand die dicken Bände mit den POSIX-Standards.**

Meine E-Mail blieb jedoch nicht unbemerkt. Jeder, der sich auskannte (und die Minix-Site wurde nur von Leuten gelesen, die sich auskannten), konnte sich ausrechnen, dass mein Projekt ein Betriebssystem sein musste. Wozu sonst sollte ich die POSIX-Regeln brauchen? Die Mail weckte die Neugier von Ari Lemke, einem Assistenten an der technischen Universität von Helsinki (wo ich studiert hätte, wenn ich nicht so an Theorie interessiert gewesen wäre). Ari schickte mir eine freundliche Antwort und bot an, mir ein Unterverzeichnis auf der ftpSite der Universität zur Verfügung zu stellen. Dort konnte ich, sobald ich so weit war, mein Betriebssystem hineinstellen, so dass jeder Interessierte es sich herunterladen konnte.

VII.

Ari Lemke muss ein ziemlicher Optimist gewesen sein. Er legte das Unterverzeichnis (ftp.funet.fi) an, lange bevor ich etwas hatte, was ich freigeben wollte. Ich besaß das Passwort, alles war vorbereitet, ich brauchte mich nur noch einzuloggen und mein Zeug rüberzuladen. Es sollte jedoch noch vier Monate dauern, ehe ich etwas hatte, das ich mit der Welt teilen wollte oder wenigstens mit Ari und den paar anderen Betriebssystemfreaks, mit denen ich E-Mails austauschte.

Ursprünglich hatte ich vorgehabt, ein Betriebssystem zu entwickeln, das ich irgendwann als Ersatz für Minix einsetzen konnte. Es musste nicht mehr leisten können als Minix, aber es musste die Sachen leisten, die mir bei Minix wichtig waren, und ein paar andere Dinge, auf die ich ebenfalls Wert legte. Beispielsweise war nicht nur die TerminalEmulation von Minix schlecht, es gab auch keine Möglichkeit, die Jobsteuerungsfunktion auszuführen - also ein Programm in den Hintergrund zu stellen, während es nicht genutzt wird. Und die Speicherverwaltung war ziemlich simpel; beim Mac-Betriebssystem ist das übrigens immer noch der Fall.'

Du entwickelst ein Betriebssystem, indem du herausfindest, was die Systemaufrufe leisten sollen, und dann dein eigenes Programm schreibst und die Systemaufrufe auf deine Weise implementierst. In Unix gibt es mehrere hundert Systemaufrufe. Einige davon können mehrere Funktionen repräsentieren. Andere sind ziemlich einfach. Einige der fundamentalen Systemaufrufe sind wirklich kompliziert und setzen das Vorhandensein einer umfangreichen Infrastruktur vo-

Anmerkung der Übersetzerin: Inzwischen ist das neue Mac-Betriebssystem OS X auf dem Markt.

raus. Denken Sie an die Systemaufrufe *Write* und *Read*. Man muss einen Plattentreiber erzeugen, um etwas auf die Platte schreiben oder von der Platte lesen zu können. Oder denken Sie an *Open*. Man muss die ganze Dateisystemsicht erzeugen, die die Namen parst und herausfindet, wo das Zeug auf der Platte liegt. Allein den Systemaufruf *Open* zu schreiben, dauerte Monate. Aber als er lief, konnte ich den gleichen Code für andere Funktionen verwenden.

Das war der Anfang. Ich las die Standards aus dem Sun-Betriebssystem-Handbuch und verschiedenen Fachbüchern heraus, nahm mir einen Systemaufruf nach dem anderen vor und versuchte, ein funktionsfähiges Programm dafür zu schreiben. Es war wirklich frustrierend.

Der Grund: Weil nichts geschieht, kannst du keinen echten Fortschritt sehen. Du kannst kleine Testprogramme schreiben, die den neu hinzugefügten Code testen. Aber das führt zu nichts. Nach einer Weile kommst du an einen Punkt, wo du keine Lust mehr hast, systematisch eine Liste von Systemaufrufen herunterzuprogrammieren. Die Sache ist so weit gediehen, dass du ein echtes Programm ausführen willst. Das erste Programm, das du ausführen musst, ist eine Shell. Bevor du eine Shell hast, ist es ziemlich schwierig, etwas anderes auszuführen. Außerdem enthält die Shell selbst viele der Systemaufrufe, die du brauchen wirst. Wenn du sie zum Laufen bringst, kannst du dir eine fortlaufende Liste der Systemaufrufe ausgeben lassen, die du noch nicht implementiert hast.

In Unix ist die Shell so etwas wie die Mutter aller Programme. Sie dient dazu, andere Binärdateien zu starten. (Eine Binärdatei ist ein Programm, das aus den Einsen und Nullen besteht, die ein Rechner liest. Immer wenn Sie ein Programm in einer Programmiersprache schreiben, kompilieren Sie anschließend den Quellcode, so dass eine Binärdatei entsteht.) Die Shell erlaubt es Ihnen überhaupt erst, sich einzuloggen. (Okay, traditionellerweise heißt das erste Programm, das man in einem echten Unix-System ausführt, *finit*, aber *finit* braucht eine umfangreiche Infrastruktur, um zu funktionieren. Es ist eine Art Controller des Geschehens. Aber wenn man im Prinzip nichts hat, was funktioniert, hat es auch keinen Sinn, *finit* zu haben.)

Statt *finit* zu starten, startete mein Kernel deshalb als Erstes die Shell. Ich hatte etwa fünfundzwanzig Systemaufrufe implementiert und die Shell war, wie gesagt, das erste richtige Programm, das ich

versuchte, zum Laufen zu bringen. Die Shell an sich hatte ich nicht selbst geschrieben. Ich hatte mir einen Klon der Bourne-Shell, einer der Original-Unix-Shells, auf eine Festplatte heruntergeladen. Sie war im Internet als Freeware zu haben, und ihr Name ging auf einen lahmen Witz zurück. Der Typ, der das Original geschrieben hatte, hieß Bourne und deshalb hieß der Klon Bourne Again Shell - wieder geborene Shell - oder abgekürzt *bash*.

Wenn du versuchst, ein echtes Programm von einer Festplatte zu laden, tritt unweigerlich ein Plattentreiber- oder Laderfehler auf, weil der Treiber oder der Lader nicht versteht, was er einliest. Deshalb gibt er einen fortlaufenden Kommentar darüber aus, was er gerade macht. Das ist wichtig, denn damit kannst du herausfinden, was falsch läuft.

Ich kam bis zu dem Punkt, wo er die Shell lud und eine Ausgabe jedes Systemaufrufs generierte, den die Shell enthielt, den ich aber noch nicht implementiert hatte. Ich bootete, führte die Shell aus und die Platte spuckte etwas in der Art wie „system call 512 is not done“ aus. Tag und Nacht starrte ich auf Ausdrücke der Systemaufrufe und versuchte herauszufinden, welche ich falsch implementiert hatte. Das machte viel mehr Spaß, als einfach nur eine Liste von Aufrufen zu implementieren. Du konntest sehen, dass etwas voranging.

Es war Ende August oder Anfang September, als ich die Shell endlich zum Laufen brachte. Von da an wurde alles leichter.

Das war eine Riesensache.

Als ich die Shell zum Laufen gebracht hatte, konnte ich fast im gleichen Atemzug noch ein paar weitere Programme kompilieren. Die Shell war komplizierter als das *cp*-Programm (um zu kopieren) oder das *ls*-Programm (um eine Liste der Verzeichnisse auszugeben). Alles, was dafür notwendig war, musste ohnehin für die Shell vorhanden sein. Sobald also die Shell funktionierte, ging es im Null-Komma-nichts von null auf 100, weil die Voraussetzungen bereits geschaffen waren. Irgendwann war so viel an Ort und Stelle, dass mit einem Mal alle Lichter gleichzeitig angingen, während vorher nichts richtig funktioniert hatte.

Ja, ich empfand ein Gefühl tiefer Zufriedenheit. Ich glaube, das war besonders wichtig, weil ich den ganzen Sommer über nichts anderes getan hatte als am Computer zu sitzen. Das ist keine Übertreibung.

Die Zeit zwischen April und August ist in Finnland praktisch die beste Zeit des Jahres. Die Leute segeln um die Inseln des Archipelago-

Nationalparks herum, liegen am Strand in der Sonne, schwitzen in den Saunas ihrer Sommerhäuser. Ich dagegen wusste kaum, ob es Tag oder Nacht, Wochentag oder Wochenende war. Meine dicken schwarzen Vorhänge sperrten die Sonne aus, die fast rund um die Uhr schien, und die Welt gleich dazu. An manchen Tagen - Nächten? - rollte ich mich direkt vom Bett in den Stuhl vor meinen Computer, der kaum einen halben Meter davon entfernt stand. Offensichtlich drängte mein Vater meine Mutter, sie sollte mich dazu bringen, einen Ferienjob anzunehmen. Aber ihr machte mein Lebensstil nichts aus: Ich störte sie nicht. Sara war ein bisschen verärgert, dass unser Telefon manchmal belegt war, wenn ich online war. Wahrscheinlich würde sie das etwas weniger diplomatisch formulieren. Ich übertreibe nicht, wenn ich sage, dass ich praktisch keinen Kontakt zur Welt außerhalb meines Computers hatte. Okay, vielleicht klopfte einmal in der Woche ein Freund an mein Fenster, und wenn ich nicht gerade wichtigen Code durchsah, bat ich ihn herein. (Es war immer ein Freund, keine Freundin - damals galten Geeks noch nicht als cool.) Wir tranken Tee und sahen uns eine Stunde oder so MTV in der winzigen Küche an. Wenn ich es mir genauer überlege, ja, doch, dann erinnere ich mich daran, dass ich gelegentlich ein Bier trinken ging oder mit zum Billard kam, wenn jemand wie Juoko lange genug ans Fenster gepocht hatte. (Ich nenne ihn „Avuton“, das heißt „Der-die-Drachen-besiegt“, aber das ist eine andere Geschichte.) Aber, ganz ehrlich, sonst passierte in meinem Leben nichts.

Und ich fühlte mich nicht im Geringsten wie ein Mitleid erregender, blasser, verquerrer Loser. Die Shell war einsatzfähig, und das bedeutete, dass ich tatsächlich den Grundstock für ein funktionsfähiges Betriebssystem gelegt hatte. Und ich hatte Spaß dabei.

Als die Shell arbeitete, begann ich, die darin eingebauten Programme zu testen. Dann kompilierte ich genügend neue Programme, um wirklich etwas damit anfangen zu können. Ich kompilierte alles unter Minix, aber ich legte die Shell in einer speziellen Partition ab, die ich für das neue Betriebssystem erzeugt hatte. Privat nannte ich es Linux.

Ehrlich: Ich wollte es nie als Linux freigeben, dafür war der Name zu selbstgefällig. Welchen Namen ich dann für eine etwaige Freigabe im Auge hatte? Freax. (Kapiert? Freaks mit dem

die das Kompilieren der Quellen beschreiben, etwa ein halbes Jahr lang das Wort „Freax“. Aber das spielte eigentlich keine Rolle. An diesem Punkt brauchte ich keinen Namen dafür, weil ich nicht vorhatte, es jemandem zugänglich zu machen.

VIII.

From: torvalds@klaava.Helsinki.Fi (Linus Benedict Torvalds) Newsgroups: comp.os.inix Subject: Welche Eigenschaften würdest du am liebsten in Minix realisiert sehen? Summary: Kleine Meinungsumfrage für mein neues Betriebssystem Message-ID: <1991Aug25.205708.9541@klaava.Helsinki.Fi> Hallo an alle, die dort draußen Minix verwenden - ich arbeite an einem (frei zugänglichen) Betriebssystem (nur so als Hobby, wird nicht groß und professionell wie GNU sein) für 386er (486er) AT-Kompatible. Die Sache ist seit April am Köcheln und nimmt allmählich Formen an. Ich hätte gern Feedback über die Dinge, die euch an Minix gefallen/nicht gefallen, da mein Betriebssystem gewisse Ähnlichkeiten dazu aufweist (unter anderem gleiche physikalische Struktur des Dateisystems (aus praktischen Gründen)). Ich habe derzeit bash (1.08) and gcc (1.40) portiert, und beides scheint zu funktionieren. Das heißt, dass ich in den nächsten Monaten etwas Einsetzbares haben werde, und ich wüsste gern, welche Features die Leute am liebsten hätten. Alle Vorschläge sind mir willkommen, aber ich verspreche nicht, dass ich sie implementieren werde :-)
Linus (torvalds@kruuna.helsinki.fi)
PS. Ja - es ist frei von Minix-Code und hat ein Multithreading-Dateisystem. Es ist nicht portierbar (verwendet 386er Taskwechsel usw.) und wird wahrscheinlich nie etwas anderes als AT-Festplatten

Die hartgesottensten Betriebssystementhusiasten in der Minix-Meute nahmen Witterung auf. Ich bekam nicht viele Vorschläge zu MinixEigenschaften, aber es gab eine Reihe anderer

>Erzähl uns mehr darüber. Braucht es eine MMU? Antwort: Ja >Wie viel davon ist in C geschrieben? Welche Schwierigkeiten werden bei der Portierung auftreten? Niemand nimmt dir ab, dass es nicht portierbar sein wird -); ich jedenfalls würde es gerne auf meinen Amiga portieren. Antwort: Es ist hauptsächlich in C programmiert, aber die meisten Leute würden das, was ich schreibe, nicht als C bezeichnen. Es verwendet jede denkbare Eigenschaft des 386ers, die ich finden konnte, weil es mir auch dazu diente, mich in den 386er einzuarbeiten. Einige meiner „C“-Dateien enthalten fast so viel Assembler wie C. Wie bereits gesagt, es verwendet eine MMU, sowohl für das Paging (noch nicht auf Festplatte) als auch für die Segmentierung. Es ist die Segmentierung, die es WIRKLICH abhängig macht vom 386er (jede Task hat ein 64 Megabyte-Segment für Code&Daten - max 64 Tasks in einem Adressraum von 4 Gigabyte. Wer mehr als 64 Megabyte/Task braucht, wird es schwer haben).

Und ein paar Leute boten sich sogar als Betatester an. Letztendlich war es keine große Entscheidung für mich, es ins Internet zu stellen. Ich war daran gewöhnt, Programme auf diesem Weg auszutauschen. Die einzige wirkliche Frage für mich war: **Wann ist der Punkt erreicht, an dem ich es den Leuten mit einem gutem Gefühl zeigen kann?** Oder genauer gesagt: Wann ist es gut genug, dass ich mich nicht dafür zu schämen brauche? Mein Ziel war es, einen Compiler und eine echte Umgebung zur Verfügung zu haben, so dass man Programme in Linux selbst erzeugen konnte, ohne Minix verwenden zu müssen. Aber als die GNU-Shell lief, war ich so stolz darauf, dass ich bereit war, es der Welt zu präsentieren. Und außerdem wollte ich Feedback bekommen.

Als die Shell funktionierte, hatte ich ein paar rudimentäre Binärdateien, die ich für das Betriebssystem kompiliert hatte. Man konnte eigentlich nichts damit anfangen, aber immerhin war erkennbar, dass es Unix ähnlich sah. Tatsächlich funktionierte es wie ein Unix mit gestutzten Flügeln.

Kurzerhand beschloss ich, es zugänglich zu machen. Ich wollte die Sache nicht an die große Glocke hängen. Stattdessen informierte ich einfach eine Hand voll Leute, vielleicht fünf oder zehn insgesamt, durch private E-Mails, dass ich es auf die ftp-Site geladen hatte. Unter ihnen waren auch Ari Lemke und Bruce Evans, der Minix-Guru. Ich stellte den Linux-Quellcode und ein paar Binärdateien in das ftp-Verzeichnis, so dass man etwas starten konnte. Ich erklärte den Leuten, wie sie das Ding zum Laufen bringen konnten. Nach wie vor mussten sie Minix installiert haben - die 386er-Version - und sie brauchten den GCC-Compiler. Genauer gesagt, sie brauchten *meine* GCC-Version. Deshalb stellte ich auch den Compiler ins Netz.

Für die Nummerierung von Freigabeversionen gibt es ein inoffizielles Protokoll. Es handelt sich um reine Psychologie. Wenn du deine Version für wirklich freigabereif hältst, gibst du ihr die Versionsnummer 1.0. Vorher aber wählst du eine Versionsnummer, aus der hervorgeht, wie viel Arbeit noch bis zu einer Version 1.0 zu leisten ist. Mit diesem Gedanken im Kopf gab ich dem Betriebssystem, das ich auf die ftp-Site stellte, die Versionsnummer 0.01. So wusste jeder, dass es weit davon entfernt war, fertig zu sein.

Ja, ich weiß noch das Datum: 17. September 1991.

Ich glaube nicht, dass mehr als ein oder zwei Leute es sich jemals richtig anschauten. Dazu hätten sie sich die Mühe machen müssen, den speziellen Compiler zu installieren, eine neue Partition anzulegen, um sie zum Booten zu nutzen, und meinen Kerneil zu kompilieren, um dann gerade mal die Shell ausführen zu können. Mehr war im Prinzip nicht möglich. Doch, man konnte den Quellcode ausdrucken, der sich auf bescheidene 10000 Zeilen beschränkte - in einem kleinen Font ausgedruckt sind das weniger als 100 Seiten auf Papier. (Heute umfasst das Ganze etwas in der Größenordnung von 10 Millionen Zeilen.)

Hauptsächlich verteilte ich das Betriebssystem, um zu demonstrieren, dass ich nicht bloß Wind gemacht, sondern wirklich etwas Greifbares vorzuweisen hatte. Im Internet redet es sich leicht. Ob es um Betriebs-

systeme geht oder um Sex, viel zu viele Leute im Cyberspace bluffen nur. Wenn du herumerzählt hast, dass du ein Betriebssystem baust, ist es deshalb schön, sagen zu können: „Schaut, ich habe tatsächlich etwas zustande gebracht. Ich habe nicht nur rumgetönt. Hier ist das Ergebnis ...“

Ach ja, Ari Lemke, der dafür gesorgt hatte, dass es seinen Weg auf die ftp-Site fand, hasste den Namen Freax. Er bevorzugte meinen anderen Arbeitsnamen, Linux, und benannte mein Posting: pubOS/Linux. Ich gebe zu, dass ich mich nicht groß dagegen zur Wehr setzte. Aber der Anstoß kam von ihm. Deshalb kann ich mit reinem oder jedenfalls einigermaßen reinem Gewissen sagen, dass der Name nicht meiner Eitelkeit entsprang. Aber ich dachte, okay, das ist ein guter Name, und ich kann seine Wahl notfalls immer jemand anderem in die Schuhe schieben, was ich hiermit tue.

Wie gesagt, mein Betriebssystem war nicht sehr nützlich. Zum einen stürzte es sehr leicht ab, wenn der Speicherplatz knapp wurde oder wenn man etwas Ungewöhnliches versuchte. Und selbst bei ganz normalem Betrieb stürzte es ab, wenn man es über längere Zeit hinweg laufen ließ. Aber in diesem Stadium war es nicht dafür gedacht, dass man es ausführte. Es war dazu gedacht, dass man es ansah. Und bewunderte.

Ich wollte meine neue Errungenschaft also lediglich den paar Leuten vorführen, die sich für die Entwicklung neuer Betriebssysteme interessierten. Ausgesprochenen Technikfreaks, die selbst unter Technikfreaks einen Spezialfall darstellten.

Ihre Reaktionen waren ausnahmslos positiv, wenn man Reaktionen wie „Es wäre schön, wenn es auch das könnte“ oder „Es sieht cool aus, aber auf meinem Computer funktioniert es überhaupt nicht“ als positiv wertet.

Ich erinnere mich an eine E-Mail, in der mir jemand schrieb, dass ihm mein Betriebssystem wirklich gefiel, und mindestens einen Absatz lang lobte, wie toll es war. Dann erklärte er mir, dass es gerade seine Festplatte aufgefressen hatte, und dass mein Plattentreiber irgendwie verrückt spielte. Seine gesamte Arbeit war verloren gegangen, aber trotzdem äußerte er sich sehr positiv. Es machte Spaß, solche E-Mails zu lesen. Aber seine war eigentlich ein Fehlerbericht über etwas, das ihm alles kaputt gemacht hatte.

Jedenfalls bekam ich genau die Art von Rückmeldungen, die ich mir erhofft hatte. Ich merzte ein paar Fehler aus, zum Beispiel den, der zum Absturz führte, wenn der Speicherplatz zu Ende ging. Und ich unternahm den großen Schritt, den GCC-Compiler auf mein Betriebssystem zu portieren, so dass ich kleine Programme direkt kompilieren konnte. Damit brauchten die Anwender nicht mehr meinen GCCCompiler zu laden, bevor sie das Betriebssystem ausführten.

IX.

„Sehnst du dich nach den Tagen zurück, wo
Männer noch Männer waren und ihre
Gerätetreiber selbst schrieben?“

Ankündigung der Freigabe der Linux-Version 0.02

Anfang Oktober erfolgte die Freigabe von Version 0.02, die mehrere Fehlerkorrekturen und einige zusätzliche Programme umfasste. Im folgenden Monat gab ich Version 0.03 frei.

Wahrscheinlich hätte ich Ende 1991 die Sache eingestellt. Ich hatte eine Menge Dinge zuwege gebracht, die ich interessant fand. Nicht alles funktionierte wirklich perfekt, aber in der Softwarewelt verliert man, sobald die fundamentalen Probleme gelöst sind, meiner Erfahrung nach leicht das Interesse. Und so ging es auch mir. Softwarefehler zu beheben ist nicht besonders spannend. Aber dann passierten zwei Dinge, die mich dazu bewogen, weiterzumachen. Erstens: Ich zerstörte aus Versehen meine Minix-Partition. Und zweitens: Ich bekam immer mehr Feedback.

Damals bootete ich in Linux, nutzte aber Minix als Hauptentwicklungsumgebung. Unter Linux las ich allenfalls E-Mails und Nachrichten vom Uni-Computer mit dem Terminal-Emulator, den ich geschrieben hatte. Der Uni-Computer war ständig belegt, und deshalb hatte ich ein Programm geschrieben, das sich automatisch selbst einwählte. Nur dass ich im Dezember statt des Modems aus Versehen meine Festplatte anwählte. Eigentlich wollte ich `/dev/tty1`, die serielle Verbindung, anwählen lassen. Aber aus Versehen wählte ich `/dev/hda1` an, die Festplatte. Das führte dazu, dass ungewollt einige der kritischsten Teile der Partition, auf der sich Minix befand, überschrieben wurden. Richtig, damit konnte ich Minix nicht mehr booten.

An diesem Punkt musste ich eine Entscheidung treffen: Ich konnte Minix neu installieren. Oder ich konnte aufs Ganze gehen und unterstellen, dass Linux mittlerweile so gut war, dass ich Minix nicht mehr brauchte. Ich würde die Programme schreiben, um Linux unter sich selbst zu kompilieren, und wenn ich zwischendurch das Gefühl hatte, doch Minix zu benötigen, würde ich einfach Linux um die gewünschte Eigenschaft ergänzen. Es ist ein riesiger konzeptueller Schritt, wenn du die ursprüngliche Host-Umgebung aufgibst und dein System völlig selbstständig betreibst, ein so großer Schritt, dass ich die neue Version Ende November als 0.10 freigab. Ein paar Wochen später folgte Version 0.11.

Und genau jetzt begannen tatsächlich ein paar Leute, es zu nutzen und etwas damit anzufangen. Bis dahin hatte ich Fehlerkorrekturen bekommen, die vielleicht eine Zeile lang waren. Aber jetzt forderten die Leute neue Eigenschaften an. Ich weiß noch, wie ich losging und meine Maschine von 4 Megabyte RAM auf 8 Megabyte aufrüstete, um meinen zusätzlichen Bedarf an Speicherplatz zu decken. Außerdem kaufte ich einen Gleitkomma-Koprozessor, weil mich verschiedene Leute gefragt hatten, ob Linux ihren Gleitkomma-Koprozessor unterstützen würde. Mit der zusätzlichen Hardware konnte mein Computer Gleitkommaberechnungen durchführen.

Ich erinnere mich, dass im Dezember dieser Typ aus Deutschland Kontakt zu mir aufnahm, der nur 2 Megabyte RAM hatte. Er versuchte, den Kernel zu kompilieren, und konnte GCC nicht ausführen, weil GCC damals mehr als ein Megabyte benötigte. Er fragte mich, ob Linux mit einem kleineren Compiler kompiliert werden konnte, der nicht so viel Speicherplatz brauchte. Also entschloss ich mich, diese Eigenschaft speziell für diesen Anwender zu realisieren, obwohl ich selbst sie im Grunde nicht brauchte. Sie heißt Page-to-Disk, und damit konnte der Typ, obwohl er nur 2 Megabyte RAM besaß, den Anschein erwecken, mehr Speicher zu haben, indem er die Platte als Speicher nutzte. Das war um Weihnachten 1991 herum. Ich erinnere mich, wie ich am 23. Dezember versuchte, die Page-to-Disk-Funktion zum Laufen zu bringen. Am 24. Dezember lief sie so recht und schlecht, stürzte aber immer wieder ab. Am 25. Dezember hatte ich es dann geschafft. Page-to-Disk war praktisch die erste Eigenschaft, die ich entwickelte, um den Bedürfnissen eines anderen gerecht zu werden. Und darauf war ich stolz.

Sie müssen nicht glauben, dass ich meiner Familie gegenüber auch nur das Geringste davon erwähnte, als wir uns bei meiner Großmutter väterlicherseits (Farmor!) trafen und uns Schinken und verschiedene Heringsgerichte schmecken ließen. Mit jedem Tag wurde die Gemeinde der Linux-Benutzer größer, und ich erhielt E-Mails aus Ländern wie Australien und den USA, die ich liebend gern besucht hätte. Fragen Sie mich nicht warum, aber ich verspürte kein Bedürfnis, mit meinen Eltern, meiner Schwester oder anderen Verwandten über die Entwicklungen in meinem Leben zu reden. Sie verstanden nichts von Computern. Wahrscheinlich dachte ich, sie würden nicht verstehen, was da eigentlich abging.

Ihnen fiel lediglich auf, dass ich mit meinem Modem ständig das Telefon blockierte. In Helsinki gab es nachts einen Billigtarif, und deshalb versuchte ich, einen Großteil der Arbeit spät abends zu Hause zu erledigen. Aber gelegentlich blockierte ich das Telefon auch tagsüber. Ich versuchte, eine zweite Leitung zu bekommen, aber das Haus, in dem die Wohnung meiner Mutter lag, war so alt, dass es keine zusätzlichen Leitungen gab, und niemand hatte Interesse daran, neue zu verlegen. Sara tat in dieser Zeit nichts anderes, als mit ihren Freunden zu telefonieren. Jedenfalls war das mein Eindruck. Darüber gerieten wir uns gelegentlich in die Haare. Virtuell. Während sie mit ihren Freunden telefonierte, ließ ich das Modem mit dem Wählen beginnen, so dass sie di-di-di-di-di hörte, wenn ich die Leitung haben wollte. Das störte sie natürlich, aber zumindest wusste sie dann, dass ich wirklich ganz dringend eine E-Mail lesen musste. Ich habe nie behauptet, der beste große Bruder der Welt zu sein.

Page-to-Disk war eine ziemlich große Sache, weil es eine Eigenschaft war, die Minix nie besessen hatte. Die Neuerung war in Version 0.12 enthalten, die ich 1992 in der ersten Januarwoche freigab. Die Leute fingen daraufhin sofort an, Linux nicht nur mit Minix, sondern auch mit Coherent zu vergleichen, einem kleinen Unix-Klon, den die Mark Williams Company entwickelt hatte. Die Erweiterung um Page-toDisk verschaffte Linux von Anfang an einen Vorsprung gegenüber der Konkurrenz.

An diesem Punkt hob Linux ab. Plötzlich stiegen die Leute von Minix auf Linux um. Damals konnte Linux noch nicht alles, was Minix konnte, aber es konnte die meisten Sachen, auf die es den Leuten ankam. Und es besaß genau die eine Fähigkeit, auf die sie wirklich

Wert legen. Mit Page-to-Disk konntest du größere Programme laufen lassen, als in deinen Speicher passten. Wenn der Speicher knapp wurde, konntest du alte Hauptspeichereinhalte nehmen, auf Platte auslagern, dir merken, wo du sie abgelegt hattest, und den Hauptspeicherbereich für das aktuell zu lösende Problem wieder verwenden. In den ersten Wochen des Jahres 1992 war das ein Riesending.

Es war im Januar, als der Kreis der Linux-Anwender von fünf, zehn, zwanzig Leuten, die ich anmailen konnte und deren Namen ich kannte, auf hunderte nicht mehr identifizierbare Leute anwuchs. Ich kannte nicht mehr jeden, der Linux einsetzte, und das machte mir Spaß.

Ungefähr zu dieser Zeit verbreitete sich im Internet in Windeseile ein übler Scherz. Ein krebskranker Junge namens Craig lag im Sterben, und ein Kettenbrief forderte die Leute auf, ihm eine Postkarte zu schicken, um ihm ihre Unterstützung zu zeigen. Die ganze Sache stellte sich als geschmackloser Witz heraus; ich glaube nicht, dass Craig je wirklich existierte, geschweige denn an Krebs erkrankt war. Aber der Aufruf generierte Millionen von Postkarten. Deshalb war es mir nur halb ernst, als ich die Leute, die Linux einsetzten, um Postkarten statt Geld bat. Das Ganze war eher als Witz gemeint, so nach dem Motto O-Gott-nicht-noch-eine-dieser-E-Mail-Bitten-um-einePostkarte. In der PC-Welt von damals gab es eine starke SharewareTradition. Man lud ein Programm herunter und ließ dem Autor etwas in der Größenordnung von zehn Dollar zukommen. Und so bekam ich E-Mails von Leuten, die wissen wollten, ob sie mir dreißig Dollar oder so schicken sollten. Ich musste ihnen irgendeine Antwort geben.

Rückblickend betrachtet wäre das Geld wohl nützlich gewesen. Ich hatte Studentenkredite in Höhe von etwa 5000 Dollar aufgenommen, und musste ungefähr 50 Dollar im Monat hinlegen, um meinen Computer abzubezahlen. Ansonsten ging mein Geld hauptsächlich für Pizza und Bier drauf. Aber Linux beschäftigte mich zu dieser Zeit so sehr, dass ich nicht viel wegging, vielleicht höchstens einmal pro Woche. Ich brauchte kein Geld für Verabredungen. Ich hätte es für Hardwarezubehör ausgeben können, aber das musste nicht sein. Ein anderer Sohn hätte wahrscheinlich Geld genommen, und sei es nur, um seiner arbeitenden allein erziehenden Mutter einen Zuschuss zur Miete rüberzuschieben. Es kam mir damals einfach nicht in den Sinn. Sie können mich ja anzeigen.

Viel interessanter fand ich es, herauszufinden, wo überall die Leute Linux nutzten. Deshalb zog ich Postkarten dem schnöden Mammon vor. Und sie strömten herein. Aus Neuseeland. Aus Japan. Aus den Niederlanden. Aus den USA. Normalerweise brachte Sara die Post herein, und musste plötzlich beeindruckt feststellen, dass ihr streitsüchtiger älterer Bruder Post von neuen Freunden aus allen Ecken der Welt bekam. Zum ersten Mal dämmerte ihr, dass ich in den vielen Stunden, in denen ich das Telefon blockierte, möglicherweise etwas Sinnvolles machte. Insgesamt waren es hunderte von Postkarten, und ich habe keine Ahnung, wo sie hingekommen sind. Sie müssen bei einem meiner Umzüge verloren gegangen sein. Avuton findet, ich sei der „am wenigsten nostalgische Mensch“, den er je getroffen hat.

Tatsächlich wollte ich das Geld aus einer ganzen Reihe von Gründen nicht. Als ich Linux erstmals ins Internet stellte, hatte ich das Gefühl, in die jahrhundertealten Fußstapfen der Wissenschaftler und Forscher zu treten, die ihre Arbeit auf den Grundfesten anderer aufsetzen - auf den Schultern von Giganten, um mit Isaac Newton zu sprechen. Ich machte meine Arbeit nicht nur zugänglich, damit andere einen Nutzen daraus ziehen konnten, ich wollte auch etwas für mich: Feedback (okay, und Anerkennung). Ich sah keinen Sinn darin, Leuten Geld abzuknöpfen, die mir möglicherweise helfen konnten, meine Arbeit zu verbessern. Vermutlich hätte ich die Sache anders angepackt, wenn ich nicht in Finnland aufgewachsen wäre, wo jeder, der das leiseste Anzeichen von Gier zeigt, misstrauisch oder sogar neidisch beäugt wird. (Das hat sich etwas verändert, seit Nokia-Handfies in den Jacken-, Hosen- und Manteltaschen der ganzen Welt zu finden sind und die Bankkonten zahlreicher Finnen anschwellen lassen.) Und, ja, ich hätte die ganze Kein-Geld-Sache zweifellos anders angepackt, wenn ich nicht unter dem Einfluss meines Großvaters, einem eingefleischten Akademiker, und meines Vaters, einem eingefleischten Kommunisten, erzogen worden wäre.

Jedenfalls wollte ich Linux nicht verkaufen. Und ich wollte die Kontrolle darüber nicht verlieren. Das heißt, ich wollte auch nicht, dass andere es verkauften. Ich machte das in der Copyright-Bestimmung zur kopierfähigen Datei der ersten Version deutlich, die ich damals im September ins Netz gestellt hatte. Dank der Berner Konvention irgendwann im 19. Jahrhundert besitzt man das Urheberrecht an allem, was man geschaffen hat, es sei denn man verkauft es. Als Urheber bestimmte ich die Regeln: Du kannst das Betriebssystem umsonst

nutzen, solange du es nicht verkaufst, und wenn du Veränderungen oder Verbesserungen vornimmst, musst du sie jedem als Quellcode zugänglich machen (nicht als Binärdateien, die unzugänglich sind). Wer diesen Regeln nicht zustimmte, hatte kein Recht, den Code zu kopieren oder damit zu arbeiten.

Versetzen Sie sich in meine Situation. Sie stecken sechs Monate Ihres Lebens in diese Sache und Sie möchten sie zugänglich machen und Sie möchten etwas davon haben, aber Sie möchten nicht, dass andere das ausnutzen. Ich wollte, dass die Leute Linux sehen und nach Herzenslust verändern und verbessern konnten. Aber gleichzeitig wollte ich sicherstellen, dass ich auch etwas davon hatte und mitbekam, was sie damit machten. Ich wollte immer Zugriff auf den Quellcode haben und Verbesserungen, die andere vornahmen, selbst nutzen können. Nach meiner Überzeugung hatte Linux die beste Chance, sich zu einem optimalen technischen System zu entwickeln, wenn ich ihm seine Reinheit bewahrte. Wenn Geld ins Spiel kam, würde die Sache undurchsichtig werden. Aber wenn du gar nicht erst zulässt, dass Geld ins Spiel kommt, hältst du dir gierige Leute vom Leib.

Während ich selbst kein Interesse daran hatte, Geld für Linux zu verlangen, zögerten andere Leute nicht, einen finanziellen Beitrag zu fordern, wenn sie jemandem eine Kopie des Betriebssystems überließen, die sie auf eine Floppydisk geladen hatten. Im Februar war es nichts Ungewöhnliches mehr, dass Leute zu Unix-User-Treffen kamen und mit Floppydisks bewaffnet waren, auf denen sich Linux befand. Die Leute fingen an, mich zu fragen, ob sie fünf Dollar oder so verlangen konnten, um wenigstens die Kosten für die Diskette und ihre Zeit abzudecken. Das Problem dabei war, dass das mein Copyright verletzte.

Es war an der Zeit, meine Linux-ist-nicht-verkäuflich-Einstellung zu überdenken. An diesem Punkt gab es bereits so viele Online-Diskussionen über Linux, dass ich ziemlich sicher war: Niemand würde es einfach nehmen und damit durchbrennen können, wie ich anfangs befürchtet hatte. Wenn jemand versuchen sollte, Linux zu stehlen und in ein kommerzielles Projekt zu verwandeln, hätte er mit dem empörten Aufschrei einer wachsenden Gemeinde von Hackern rechnen müssen, die klar gestellt hätten: „Hey, das ist Linux! Das können Sie nicht machen“, wenn auch in weniger höflichen Worten.

Linux hatte eine Eigendynamik entwickelt. Täglich tauschten sich Hacker auf der ganzen Welt über Änderungsvorschläge aus. Gemein-

sam schufen wir das beste Betriebssystem, das es gab, und es bestand eigentlich keine Gefahr, dass wir von diesem Pfad abkommen würden. Deswegen, und weil Linux einen so hohen Wiedererkennungsgrad hatte, legten sich meine Bedenken, den Leuten seinen Verkauf zu erlauben.

Bevor Sie mich jetzt für Mr. Wohltäter halten, möchte ich Ihnen ein weiteres wichtiges Kriterium für meine Entscheidung nennen. Tatsache ist: Um Linux nutzbar zu machen, hatte ich mich auf eine Menge Tools verlassen, die frei über das Internet verteilt worden waren - ich hatte mich auf die Schultern von Giganten gehievt. Das wichtigste dieser frei zugänglichen Programme war der GCC-Compiler gewesen. Das Urheberrecht an ihm wurde nach der General Public License geschützt, die weltweit als die GPL (oder das „Copyleft“) bekannt und ein geistiges Produkt von Richard Stallman ist. Nach den Begriffen der GPL ist Geld nicht das Thema. Du kannst eine Million Dollar verlangen, wenn du jemanden findest, der sie dir zahlt. Aber du musst den Quellcode verfügbar machen. Und die Person, der du den Quellcode gibst oder verkaufst, muss alle Rechte besitzen, die auch du besitzt. Es ist ein brillantes Konzept. Aber im Gegensatz zu vielen hartgesottenen GPL-Freaks, die sich auf den Standpunkt stellen, jede Softwareinnovation solle unter der General Public License der Welt zugänglich sein, glaube ich, dass es den individuellen Erfindern selbst überlassen bleiben sollte, zu entscheiden, was sie mit ihrer Erfindung machen wollen.

Deshalb verwarf ich mein altes Copyright und übernahm die GPL, ein Dokument, das Stallman verfasst hatte und von Anwälten hatte überprüfen lassen. (Weil Anwälte daran beteiligt waren, erstreckt es sich über mehrere Seiten.)

Das neue Copyright war in Version 0.12 enthalten, aber ich weiß noch, wie ich in der Nacht nach der Freigabe wach lag und nervös überlegte, wie sich kommerzielle Interessen auf das System auswirken würden. Rückblickend klingt es lächerlich angesichts der Tatsache, wie klein das kommerzielle Interesse war. Aber irgendwie dachte ich, ich müsste vorsichtig sein. Eine meiner Sorgen war - und ist - es, dass jemand Linux einfach nimmt und die Urheberrechte nicht anerkennt. Damals machte ich mir Sorgen, dass ich gegen eine Copyright-Verletzung in den USA praktisch nicht einmal Klage erheben konnte. Das stellt bis heute ein Problem für mich dar. Es ist zwar einfach, jemanden

wegen einer Copyright-Verletzung zu belangen, aber ich mache mir Sorgen, was in der _ Zeit geschieht, bevor der Sache Einhalt geboten werden kann.

Außerdem plagt mich die Angst, dass Unternehmen in Ländern wie China die GPL nicht einhalten. Praktisch nichts in ihren Rechtssystemen hält solche Länder davon ab, das Urheberrecht zu verletzen, und von einer höheren Warte aus betrachtet lohnt es sich nicht, Leute zu verfolgen, die sich illegal verhalten. **Große Softwareunternehmen und die Musikindustrie haben es versucht - mit mäßigem Erfolg.** Die Realität nimmt meinen Ängsten die Spitze. **Auch wenn jemand Linux eine Weile lang illegal nutzt, werden die Leute, die das Copyright einhalten und Veränderungen am Kerne zurückmelden und damit zu seiner Verbesserung beitragen, langfristig am längeren Hebel sitzen.** Sie sind ein Teil des Prozesses und nehmen an **der Aktualisierung des Kernels** teil. Leute, **die die GPL nicht einhalten, müssen dagegen auf Upgrades verzichten.** Ihre Kunden werden sich von ihnen abwenden. Hoffe ich.

Grundsätzlich haben Copyright-Bestimmungen für mich zwei Seiten. Angenommen, jemand verdient im Monat 50 Dollar. Darf man von ihm wirklich erwarten, 250 Dollar für Software zu zahlen? **Ich finde nicht, dass er unmoralisch handelt, wenn er die Software verbotenerweise kopiert und für die gesparten fünf Monatsgehälter Lebensmittel einkauft.** Diese Art von Copyright-Verletzung ist moralisch betrachtet in Ordnung. **Und es ist unmoralisch - und dumm obendrein - den „Gesetzesbrecher“ zu verfolgen.** Um auf Linux zurückzukommen: Wen interessiert es schon, wenn ein Einzelner die GPL nicht wirklich einhält, solange **er Linux nur für seine eigenen Zwecke nutzt?** Unmoralisch finde ich es, wenn jemand sich seiner bemächtigt, um das schnelle Geld zu machen, ganz gleich, ob er es in den USA oder Afrika tut. Aber selbst dann ist es eine Frage der Größenordnung. Gier ist nie gut.

X.

Minix gegen Linux

Aber die Resonanz war nicht nur positiv. Obwohl das Austragen von Konflikten noch nie zu meinen Stärken zählte, sah ich mich gezwungen, Linux und meine Männlichkeit zu verteidigen, als Andrew Tanenbaum das Betriebssystem, das sein eigenes zu ersetzen begann, zunehmend unter Beschuss nahm. Wir sind Nerds, deshalb fanden unsere Auseinandersetzungen per E-Mail statt.

Wer konnte ihm einen Vorwurf daraus machen, dass er unter seinem T-Shirt ins Schwitzen geriet? **Bevor die ersten Linux-Newsgroups entstanden waren, nutzte ich regelmäßig die Minix-Newsgruppen, um**

Ankündigungen über Linux zu machen oder Leute zu finden, die an dem Betriebssystem Interesse hatten. Was hätte Andrew daran gefallen sollen?

So sah er es von Anfang an nicht gerne, dass ich in seiner Newsgroup mitmischte. Und zweifellos **war er nicht sonderlich begeistert davon, dass diese neue Kreation aus der schneereichen finnischen Wildnis anfang, sein Betriebssystem in den Schatten zu stellen - und dass das Projekt so viele Entwickler anzog.** Er hatte auch andere Vorstellungen über den Aufbau von Betriebssystemen als ich. Damals gehörte Andrew einem Lager von Informatikern an, die den Microkernel-Ansatz für Betriebssysteme bevorzugten. Er **hatte Minix als Microkernel programmiert, und Amoeba, das System, an dem er damals arbeitete, verwendete ebenfalls einen Microkernel.**

Diese Richtung erfreute sich Ende der 1980er-, Anfang der **1990erJahre besonderer Beliebtheit und sah sich durch den Erfolg von Linux bedroht.** Deshalb stellte Andrew unfreundliche Sticheleien gegen Linux ins Netz.

Hinter dem Microkernel steht die Theorie, dass Betriebssysteme kompliziert sind. Deshalb versuchst du, einen Teil der Komplexität zu beseitigen, indem du einen Großteil des Betriebssystems modularisierst. Zu den Lehrsätzen des Microkernel-Ansatzes gehört es, dass der Kernei, der Kern des Kerns des Kerns, so wenig tun soll wie irgend möglich. Seine Hauptfunktion ist die Kommunikation. All die verschiedenen Funktionen des Computers sind Dienstleistungen, auf die man über die Kommunikationskanäle des Microkernel zugreifen kann. Es gilt, den Problemraum so stark zu unterteilen, dass keine Komplexität mehr übrig bleibt.

In meinen Augen war das Blödsinn. Es stimmt, jedes Teil für sich allein ist einfach. Aber durch die Interaktionen wird die Sache weitaus komplizierter, als wenn viele der Dienstleistungen wie bei Linux gleich im Kernei selbst enthalten sind. Denken Sie nur an Ihr Gehirn. Jedes Teil für sich allein ist einfach. Aber die Interaktionen zwischen den Teilen Ihres Gehirns verkomplizieren das System ganz ungemein. Es ist das alte Das-Ganze-ist-größer-als-die-Teile-Problem. Wenn du das Problem in zwei Hälften teilst und zugrundelegst, dass die Hälften jeweils nur halb so kompliziert sind wie das Ganze, ignorierst du die Tatsache, dass sich der Grad der Komplexität um die notwendige Kommunikation der beiden Hälften erhöht. Der Microkernel basiert auf der Theorie, dass du den Kernei in 50 unabhängige Teile zer schlägst, von denen jeder nur ein Fünfzigstel der Komplexität des Ganzen besitzt. Aber niemand schien sich klar zu machen, dass die Kommunikation zwischen den Teilen noch komplizierter ist als das ursprüngliche System - ganz zu schweigen davon, dass die Teile auch weiterhin nicht trivial sind.

Das ist das Hauptargument gegen Microkernels. Die Einfachheit, die du zu erreichen suchst, ist trügerisch.

Linux begann viel kleiner und viel, viel einfacher. Es erzwang keine Modularität. Deshalb konnte man viele Dinge viel direkter lösen als mit Minix. Schon ganz am Anfang war mir an Minix ein Problem aufgefallen: Wenn fünf verschiedene Programme gleichzeitig liefen, die jeweils eine andere Datei lesen wollten, mussten die Tasks serialisiert werden. In anderen Worten: Es konnte vorkommen, dass fünf verschiedene Prozesse Anforderungen an das Dateisystem schickten: „Kann ich bitte aus Datei X lesen?“ In diesem Fall pflegte der Dateisystem-Dämon, der für das Lesen zuständig ist, die erste

zu nehmen und zurückzusenden, dann die zweite zu nehmen und zurückzusenden, usw.

Unter Linux mit seinem monolithischen Kern startet dagegen jeder der fünf verschiedenen Prozesse einen Systemaufruf an den Kernei. Der Kernei muss gut darauf achten, dass keine Verwirrung entsteht, aber er ist in der Lage, ganz selbstverständlich einer beliebigen Zahl von Prozessen gerecht zu werden und ihren Anforderungen nachzukommen. Auf diese Weise ist Linux viel schneller und

Ein weiteres Problem von Minix bestand darin, dass man zwar auf den Quellcode zugreifen konnte, die Lizenzen aber keine großen Eingriffe zuließen. Nehmen Sie jemanden wie Bruce Evans, der Minix grundlegend überarbeitete und seine Nutzbarkeit enorm steigerte. Er konnte seine Verbesserungen nicht einfach einbinden, sondern war darauf beschränkt, Patches vorzunehmen. Vom praktischen Standpunkt aus betrachtet ist diese Flickschusterei eine Katastrophe. Es war ihm von Rechts wegen nicht erlaubt, den Leuten ein bootfähiges Image zur Verfügung zu stellen, so dass sie das System leicht aktualisieren konnten. Deshalb mussten die Anwender ein mehrschrittiges Verfahren durchlaufen, um das System überhaupt nutzen zu können, und das war natürlich ein Unding.

Das einzige Mal, wo ich wirklich mit Andrew Tanenbaum kommunizierte, war Anfang 1992. Stellen Sie sich vor, Sie loggen sich eines Morgens ein, die Schneestürme fegen um Ihr Haus und

From: ast@cs.vu.nl (Andy Tanenbaum)
Newsgroups: comp.os.minix
Subject: LINUX ist von gestern
Date: 29 Jan 92 12:12:50 GMT

Ich war in den letzten Wochen in den USA, deshalb habe ich nicht viel zu Linux gesagt (nicht, dass ich viel gesagt hätte, wenn ich da gewesen wäre); aber jetzt würde ich gern ein paar Dinge dazu anmerken.

Wie die meisten von euch wissen, ist MINIX für mich ein Hobby, eine Beschäftigung für die Abende, an denen ich keine Lust mehr habe, Bücher zu schreiben, und an denen CNN keine größeren Kriege, Revolutionen oder Senatsanhörungen live überträgt. Hauptberuflich bin ich Professor und forsche im Bereich Betriebssysteme.

Aufgrund meiner Beschäftigung mit dem Thema glaube ich, in etwa zu wissen, in welche Richtung sich Betriebssysteme im nächsten Jahrzehnt oder so entwickeln werden. Zwei Aspekte ragen heraus:

1. MICROKERNEL VS MONOLITHISCHES SYSTEM Die meisten älteren Systeme sind monolithisch, das heißt, das ganze Betriebssystem ist eine einzige a.out-Datei, die im „Kernelmodus“ arbeitet. Diese Binärdatei enthält die Prozessverwaltung, die Speicherverwaltung, das Dateisystem und den ganzen Rest. Beispiele für solche Systeme sind UNIX, MS-DOS, VMS, MVS, OS/360, MULTICS und viele andere.

Die Alternative dazu ist ein Microkernel-basiertes System, bei dem der Großteil des Betriebssystems als separate Prozesse läuft, im Wesentlichen außerhalb des Kernel. Sie kommunizieren durch Message-Passing. Der Kernel hat die Aufgabe, das Message-Passing zu handhaben, die Unterbrechungsbehandlung, die maschinennahe Prozessverwaltung und möglicherweise die E/A. Beispiele für diesen Entwurfsansatz sind RC4000, Amoeba, Chorus, Mach und das noch nicht frei gegebene Windows/NT.

Ich könnte an dieser Stelle zwar ausführlich auf die relativen Vorzüge der beiden Entwürfe eingehen, aber es genügt wohl, wenn ich sage, dass die Diskussion für die Leute, die wirklich Betriebssysteme entwickeln, mehr oder weniger beendet ist. Microkernels haben gesiegt.

MINIX ist ein Microkernel-basiertes System. Das Dateisystem und die Speicherverwaltung sind separate Prozesse, die außerhalb des Kernels laufen.

z. PORTABILITÄT MINIX wurde im Hinblick auf seine Portierbarkeit designt und von der Intel-Familie auf den 680x0er (Atari, Amiga, Macintosh), SPARC und den NS32016 portiert. LINUX ist ziemlich eng an den 80x86er gebunden. Das ist nicht der richtige Weg.

Versteht mich nicht falsch, ich habe nichts gegen LINUX. Ich werde damit alle Leute los, die aus MINIX ein BSD UNIX machen wollen. Aber ganz ehrlich, ich würde den Leuten, die auf ein **MODERNES** **frei** zugängliches Betriebssystem Wert legen, raten, sich

nach einem Microkernel-basierten, portierbaren Betriebssystem wie GNU oder etwas Ähnlichem umzusehen. Andy Tanenbaum (ast@cs.vu.nl)

Ich wusste, jetzt musste ich meine Ehre verteidigen, deshalb schrieb ich zurück:

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Subject: Re: LINUX ist von gestern
Date: 29 Jan 92 23:14:26 GMT
Organization: University of Helsinki

Ich fürchte, bei diesem Thema komme ich um eine Antwort nicht herum. Minix-Anwender, die ohnehin schon genug von Linux gehört haben, mögen mir verzeihen. Ich würde den **Köder** lieber einfach ignorieren, aber ... die Zeit für einen energischen Schlagabtausch ist gekommen.

Im Artikel <12595@star.cs.vu.nl> ast@cs.vu.nl (Andy Tanenbaum) schreibt Andrew Tanenbaum:

>Ich war in den letzten Wochen in den USA, deshalb >habe ich nicht viel zu Linux gesagt (nicht, dass ich >viel gesagt hätte, wenn ich da gewesen wäre); aber >jetzt würde ich gern ein paar Dinge dazu anmerken.

>Wie die meisten von euch wissen, ist MINIX für mich >ein Hobby, eine Beschäftigung für die Abende, an >denen ich keine Lust mehr habe, Bücher zu schreiben, >und an denen CNN keine >größeren Kriege, Revolutionen oder Senatsanhörungen live überträgt. Hauptberuflich bin >ich Professor und forsche im Bereich >Betriebssysteme.

Ist das Ihre Entschuldigung für die Beschränktheit von Minix? Sorry, aber damit kommen Sie nicht durch: Ich habe mehr Entschuldigungen als Sie, und trotzdem lässt Linux Minix in fast allen Bereichen alt aussehen. Ganz zu schweigen von der Tatsache, dass ein Großteil des brauchbaren Minix-Codes von Bruce Evans zu stammen scheint.

Re 1: Minix ist für Sie ein Hobby - schauen Sie sich doch an, wer mit Minix Geld macht und wer Linux frei

verteilt. Dann können Sie über Hobbies reden. Machen Sie Minix frei zugänglich, und es gäbe einen entscheidenden Grund weniger, daran herumzumeckern. Linux ist für mich immer ein echtes Hobby gewesen (aber eines, das ich ernst nehme, die beste Art von Hobby, die man sich denken kann): Ich bekomme kein Geld dafür, und es hat nicht einmal etwas mit meinem Studium zu tun. Ich habe alles in meiner Freizeit und auf meinem eigenen Rechner gemacht.

Re 2: Sie sind hauptberuflich Professor und Wissenschaftler: Das ist eine verdammt gute Entschuldigung, warum Minix so hirngeschädigt ist. Ich kann nur hoffen (und annehmen), dass Amoeba nicht genauso schrottig wird wie Minix.

>1. MICROKERNEL VS MONOLITHISCHES SYSTEM Stimmt, Linux ist monolithisch, und ich finde auch, dass Microkernels schöner sind. Bei einem weniger angriffslustigen Gesprächspartner würde ich wahrscheinlich einem Großteil des Gesagten zustimmen. Vom theoretischen (und ästhetischen) Standpunkt aus gesehen zieht Linux den Kürzeren. Wenn der GNU-Kernel im letzten Frühjahr fertig gewesen wäre, hätte ich mir nicht die Mühe gemacht, mein Projekt auch nur zu starten. Aber er war es nicht und ist es nicht. Linux gewinnt eine Menge Punkte, weil es jetzt verfügbar ist.

>MINIX ist ein Microkernel-basiertes System. [gelöscht, aber ohne den Sinn zu entstellen]. LINUX ist ein System im monolithischen Stil.

Wenn das das einzige Kriterium für die Qualität eines Kernels wäre, hätten Sie recht. Aber Sie erwähnen nicht, dass Minix den Microkernel-Anspruch nicht sehr gut erfüllt, und Probleme mit echtem Multitasking (im Kernel) hat. Wenn ich ein Betriebssystem entwickelt hätte, das Probleme mit einem Multithreading-Dateisystem hat, würde ich andere nicht so vorschnell verurteilen, sondern alles dafür tun, dieses Fiasko in Vergessenheit geraten zu lassen.

[Ja, ich weiß, dass es Multithreading-Hacks für Minix gibt, aber sie sind Hacks, und Bruce Evans sagt mir, dass es eine Menge Race-Conditions gibt.]

>2. PORTABILITÄT „Portabilität ist etwas für Leute, die keine neuen Programme schreiben können.“

Tatsache ist, dass Linux leichter portierbar ist als Minix. Was? höre ich Sie sagen. Es stimmt - aber nicht in dem Sinn, in dem ast es meint: Ich habe Linux so gut ich konnte an Standards angepasst (ohne einen POSIX-Standard vor mir liegen zu haben). Im Allgemeinen ist es /viel/ einfacher, etwas nach Linux zu Portieren als etwas nach Minix zu Portieren.

Ich räume ein, dass Portabilität eine feine Sache ist: aber nur, wo sie wirklich Sinn macht. Es ist wenig sinnvoll, ein Betriebssystem übertrieben portabel zu machen: Die Einhaltung einer portierbaren API genügt. Der eigentliche Sinn eines Betriebssystems ist es ja, die Hardware-Eigenschaften zu nutzen und hinter einer Schicht von High-Level-Aufrufen zu verbergen. Genau das leistet Linux: Es verwendet einfach eine größere Teilmenge der 386er-Eigenschaften als andere Kernels es zu tun scheinen. Natürlich führt das dazu, dass der eigentliche Kernel nicht portiert werden kann, aber dafür resultiert daraus ein /wesentlich/ einfacherer Entwurf. Das ist ein akzeptabler Kompromiss, der Linux überhaupt erst möglich machte.

Ich gebe auch zu, dass Linux die Nicht-Portabilität auf die Spitze treibt: Ich habe meinen 386er im letzten Januar bekommen und Linux unter anderem in Angriff genommen, um ihn besser kennen zu lernen. Vieles hätte portierbarer sein müssen, wenn Linux ein echtes Projekt gewesen wäre. Aber ich will nicht übertrieben viele Entschuldigungen anführen: Es war eine Entwurfsentscheidung, und als ich die Sache im letzten April begonnen habe, wusste ich noch nicht, dass jemand es wirklich würde einsetzen wollen. Ich bin glücklich, vermelden zu können, dass ich mich getäuscht habe, und da mein Quellcode frei zugänglich ist, steht es jedem frei, ihn zu Portieren, auch wenn das nicht einfach sein dürfte.

Linus

PS> Es tut mir Leid, wenn ich manchmal zu schroff klinge: Minix ist ganz gut, wenn man nichts anderes hat. Amoeba mag gut sein, wenn man 5 bis 10 übrige 386er herumliegen hat, aber ich habe sie nicht. Ich bin normalerweise nicht aufbrausend, aber wenn es um Linux geht, bin ich empfindlich :)

- ich, momentan (ironisch gemeint)

Im Zusammenhang mit diesem meinem einzigen Schlagabtausch gab es noch ein paar Fortsetzungen. Um es auf den Punkt zu bringen: Es waren Gegenstimmen da, selbst in jenen frühen Tagen. (Außerdem wichtig: Nehmen Sie sich in Acht, was Sie in einem elektronischen Forum von sich geben. Ihre Tipp- und Grammatikfehler werden Sie ein Leben lang verfolgen.)

Linus und ich lassen unsere Familien und Freunde am Zeltplatz zurück, um einen Machmittag lang an einem klaren Bach entlang zu wandern. Es ist das Wochenende des 4. Juli, und wir zelten in Grover Hot Springs, hoch oben in der östlichen Sierra, an einem Platz, der den Seiten des National Geographic entnommen sein könnte. „Das ist ein Kodak-Moment“, erklärt Linus und lässt seinen Blick über eine von wilden Blumen übersäte Wiese und die dramatischen Felswände im Hintergrund schweifen. Wir lassen uns an einem Platz neben dem Fluss nieder, und ich bitte ihn, sein Leben in jenen Tagen zu beschreiben, als die Kunde der Anziehungskraft von Linux sich weit über die ursprüngliche Fangemeinde der Mewsgroup-Enthusiasten hinaus, von denen Linus nur wenige überhaupt persönlich kennt, zu verbreiten begann.

„Das muss doch ein super Gefühl gewesen sein“, sage ich. „Jahrelang hast du dich in deinem Zimmer fast ohne Kontakt zur Welt jenseits deiner CPU abgeschuftet. Und plötzlich erkennen Menschen in jeder Ecke des Planeten die phantastische Arbeit an, die du geleistet hast. Du bist das Zentrum dieser wachsenden Gemeinde, die auf dich blickt, um...“

„Ich kann mich eigentlich nicht erinnern, dass es eine große Sache für mich gewesen wäre“, antwortet er. „Wirklich nicht. Ich hatte zwar kaum etwas anderes im Kopf, aber das lag daran, dass es ständig irgendein Problem zu lösen gab. Deshalb hat es mich beschäftigt, aber nicht, weil es emotional so wichtig für mich war. Es war vor allem eine intellektuelle Herausforderung.“

Ich freute mich darüber, dass mich so viele Leute motivierten, das Projekt fortzuführen. Ich dachte, ich hätte das Ende der Fahnenstange erreicht, einen Punkt, an dem es so gut wie fertig war. Aber dieser Punkt kam nie, weil die Leute mir immer neue Gründe gaben, weiter zu machen, und mich mit immer neuen Denksportaufgaben versorgten, über die ich nachgrübeln konnte. Das hielt mein Interesse wach.

Anderenfalls hätte ich mich wahrscheinlich einfach einem anderen Projekt zugewandt, denn so habe ich getickt, und das machte mir Spaß. Aber ich nehme an, mehr Sorgen als über Linux habe ich mir über meine Nase oder so gemacht", sagt er.

Ein paar Wochen später sind wir im Stanford Shopping Center, wo Linus ratlos vor einer Riesenauswahl von Jogging-Schuhen steht. „Wie viele Meilen laufen Sie typischerweise in der Woche? ", fragt der Verkäufer. Linus lächelt: Er ist in den letzten zehn Jahren keine einzige Meile gelaufen. Körperliche Bewegung war keine Top-Priorität für ihn gewesen. Aber in seinen schwächeren Momenten gibt Linus zu, dass er gerne ein paar überflüssige Pfunde loswerden möchte.

„Tove muss dich überredet haben, mir beim Abbau meines Rettungsringes zu helfen ", witzelt er und tätschelt seinen Bauch.

„Sag ihr, dass ihr Scheck für diese Woche noch nicht eingegangen ist", antworte ich.

Bald darauf umkreisen wir den Stanford-Campus auf der Suche nach einem rechtmäßigen Parkplatz. Ungefähr eine halbe Stunde später machen wir ein paar Stretching-Übungen und dann laufen wir über die schmalen Wege am eingetrockneten See des Campus vorbei auf unser Ziel im Wald zu: die riesige Satellitenschüssel am Hang. Aber so weit kommen wir nicht. Ich gebe ein unfair schnelles Tempo vor und bin überrascht, dass Linus sich etwa eine Meile lang dicht hinter mir halten kann. Dann gerät er außer Atem. Ein paar Minuten später lassen wir uns ins Gras neben dem See fallen.

„Wie hat deine Familie auf die Ereignisse um Linux herum reagiert?", frage ich. „Das muss doch sehr aufregend für sie gewesen sein."

„Ich glaube nicht, dass sie es wirklich mitbekommen haben", antwortet er.

„Ich will nicht sagen, dass sie sich nicht dafür interessiert hätten. Aber ich habe mein ganzes Leben lang programmiert, das war für sie nichts Neues. "

„Aber du musst deiner Familie doch etwas gesagt haben. Wenn dich dein Vater zum Beispiel irgendwohin gefahren hat, hast du dann nicht gesagt: ‚Hey, du wirst es nicht glauben, aber ich habe da auf meinem Computer etwas gemacht, was hunderte von Leuten einsetzen ...

„Nein", sagt er. „Ich hatte kein Bedürfnis, mit meiner Familie oder mit Freunden darüber zu reden. Ich wollte nicht damit hausieren gehen. Ich erinnere mich, dass Lars Wirzenius sich etwa zu der Zeit, als ich Linux schrieb, dafür entschied, XENIX, die SCO-Version von Unix, zu kaufen, und sich deswegen zu rechtfertigen versuchte, so nach dem Motto: ‚Du darfst das nicht falsch verstehen.‘ Ich glaube nicht, dass mir das etwas ausgemacht hat. Später stieg er dann um, aber das berührte mich ebenso wenig. Ich freute mich darüber, dass die Leute Linux einsetzten, und es war super, ihre Kommentare zu bekommen, aber gleichzeitig war mir das alles nicht sonderlich wichtig. Ich wollte ja nicht das Evangelium predigen. Ich war stolz darauf, dass die Leute meinen Code nutzten, aber ich kann mich nicht daran erinnern, dass ich je darüber hätte reden wollen. Und ich habe Linux nie für das Wichtigste auf der Welt gehalten. Ich dachte nicht, dass ich etwas wirklich Wichtiges tat, nur weil ein paar hundert Leute meine Software einsetzten. Für mich war das Ganze eher ein Spaß. Und so empfinde ich es immer noch."

„Du wolltest also nicht einmal mit deinen Eltern und deiner Familie und deinen Freunden darüber reden? Und was da um dich herum passierte, war wirklich nicht aufregend für dich?", vergewissere ich mich, ohne meine Ungläubigkeit zu verbergen.

Er wartet ein paar Sekunden, ehe er antwortet: „Ich erinnere mich nicht daran, ob ich damals überhaupt Gefühle gehabt habe."

Linus kauft ein neues Auto, einen BMW Z3, ein zweisitziges Cabrio, das für ihn den Inbegriff von ‚Spaß‘ bedeutet. Der Wagen ist metallic blau und sieht wie das Spielzeugauto eines wohlerzogenen Jungen aus. Er wählte die Farbe, weil der Wagen nicht in leuchtend gelb, seiner Lieblingsfarbe, zu haben war. Das BMW-Gelb, erklärt er, „sieht wie Pisse aus". Jahrelang hatte er seinen Pontiac so nahe wie möglich am Eingang zur Transmeta-Zentrale in einem Büropark von Santa Clara geparkt. Der BMW aber ist vor seinem Bürofenster abgestellt, angeblich, damit er im Schatten steht. So kann Linus gleichzeitig am Computer sitzen und sein neues Auto bewundern.

Vor etwas über einem Jahr hatten wir unseren ersten gemeinsamen Ausflug über den Berg nach Santa Cruz in einem Cabrio unternommen, einem weißen Mustang, den ich für diesen Anlass gemietet hatte. Während dieses Ausflugs hatte Linus großen Wert darauf gelegt, die

Sportwagen zu begutachten, die vor der Sauna und der Brauerei, die wir anschließend betuchten, geparkt waren. Nun fahren wir in seinem eigenen Sportwagen über den Berg. Er lächelt, als er die Kurven der Route 17 nimmt.

„Du hast ihn dir verdient“, sage ich.

Ich ziehe eine Hand voll CDs aus dem Handschuhfach.

„Pink Floyd?“, frage ich. „The Who? Janis Joplin?“

„Das ist die Musik, mit der ich aufwuchs. Ich habe mir als Jugendlicher nie CDs gekauft, aber das war die Musik, die bei uns in der Wohnung herumlag. Ich nehme an, meine Mutter hörte sie sich an, obwohl ich noch weiß, dass sie auf Elvis Costello stand.“

Es ist Freitagnachmittag, ein funkelnder Freitagnachmittag von kalifornischer Perfektion, der alle fünf Sinne entzückt: ein kobaltblauer Himmel für die Augen, hei/3e Sonnenstrahlen auf der Haut, der Duft des Bergeukalyptus, der süße Geschmack der reinen Luft, die einlullende Musik von Pink Floyd aus der hochgerüsteten Lautsprecheranlage. Wir sprühen uns mit Sun-Blocker ein und singen die Texte der Rock-Klassiker mit. Vermutlich wirken wir auf überholende Autofahrer wie schlechte Kopien zweier Postpubertierender, aber Linus' neuer BMW Z3 wird nicht oft überholt.

Wir parken zwischen schübigeren Autos neben dem Highway 1 etwas nördlich von Santa Cruz, und suchen uns einen Weg hinunter zu einem fast leeren Strand. Dort breiten wir unsere Handtücher in der warmen Sonne aus und lassen ein paar Minuten verstreichen, ehe ich meinen Kassettenrecorder aus meinem Rucksack herausnehme. Wie schon so oft fordere ich ihn auf, über den Linus jener frühen Tage zu reden.

Er malt ein Rechteck in den Sand, das sein Zimmer darstellt, und zeigt mir dann, wo das Bett und der Computer standen. „Mich aus dem Bett zu rollen und nach meiner E-Mail zu schauen, war eins“, sagt er, während er die Bewegung mit dem Finger nachzeichnet. „An manchen Tagen habe ich die Wohnung überhaupt nicht verlassen. Ich habe nicht nur nach meiner E-Mail geschaut, um zu sehen, wer mir eine Mail geschickt hat. Ich wollte vor allem wissen, ob ein bestimmtes Problem gelöst worden war. Welche neue aufregende Herausforderung haben wir heute? Oder, wenn wir ein Problem hatten, wer hat eine Lösung gefunden?“

Linus erzählt mir, dass sein gesellschaftliches Leben zu dieser Zeit „jämmerlich“ war. Dann denkt er wohl, dass das allzu jämmerlich klingt, und verbessert sich: „Sagen wir, es war einigermaßen jämmerlich.“

„Ich bin zwar nicht völlig zum Einsiedler geworden“, sagt er, „aber trotz des Wirbels um Linux war ich so ungesellig wie vorher. Du weißt, dass ich mich nie telefonisch melde. So war es schon immer. Ich rufe nie an. Die meisten meiner Freunde melden sich von sich aus. Ich nicht. Du kannst dir vorstellen, wie dein Liebesleben aussieht, wenn du eine Frau nie anrufst. Deshalb beschränkte sich mein gesellschaftliches Leben zu jener Zeit auf ein paar Freunde, die an mein Fenster klopfen, wenn sie auf eine Tasse Tee hereinkommen wollten. Ich glaube nicht, dass damals jemand den Unterschied bemerkte - Oh, er arbeitet da an einer wirklich großen, wichtigen Sache und wird eines Tages die Welt verändern. Ich glaube nicht, dass irgendjemand etwas in der Art dachte.“

Linus' Gesellschaftsleben erschöpfte sich in dem wöchentlichen Spektrum-Abend, wo er andere Studenten der Naturwissenschaften traf. Dieses soziale Event bereitete ihm weitaus mehr Kopfzerbrechen als jede technische Herausforderung.

„Wovor ich Angst hatte? Einfach vor Sozialkontakten im Allgemeinen. Vielleicht ist ‚Angst‘ das falsche Wort, aber es gab da schon eine recht starke emotionale Komponente. Allein schon der Gedanke an Mädchen. Linux war für mich damals nicht so wahnsinnig wichtig. Und bis zu einem gewissen Grad ist es das auch heute nicht. Bis zu einem gewissen Grad kann ich es immer noch ignorieren.“

In diesen ersten Jahren an der Uni war mir die soziale Seite wirklich wichtig. Es war nicht so, als hätte ich mich wegen meines Rundrückens geschämt und dass die Leute darüber lachen könnten. Es war eher so, dass ich Freunde haben wollte und alles, was dazugehört. Ich ging vor allem deshalb so gern zu den Spektrum-Treffen, weil sie dir einen Rahmen boten, Sozialkontakte zu pflegen, ohne dass du kontaktfreudig sein musstest. An diesem einen Abend war ich kontaktfreudig, und an allen anderen Abenden hockte ich vor dem Computer. Es war für mich etwas viel Emotionaleres als Linux. Linux hat mich nie aufgewühlt. Ich habe wegen Linux nie auch nur eine schlaflose Nacht verbracht.

Was mich wirklich aufwühlte und immer noch aufwühlt, ist nicht die Technik an sich, sondern die soziale Interaktion, die damit verbunden ist. Es waren weniger die technischen Fragen, die mich auf das Posting von Andrew Tanenbaum so heftig reagieren ließen. Hätte es ein anderer geschrieben, hätte ich mich kurz aufgeregt und die Sache vergessen. Das Problem war, dass er seine Nachricht über die Mailing-List verbreitete und mich ... Das soziale Ansehen, das ich bei diesen Leuten genoss, war mir wichtig, und er griff dieses Ansehen an.

Eines der Dinge, die ich an Linux so schön und motivierend fand, war das Feedback, das ich bekam. Es bedeutete, dass Linux wichtig war, und war ein Zeichen, dass ich einer sozialen Gruppe angehörte. Die ich anführte. Das war mir wichtig, keine Frage, viel wichtiger, als meinen Eltern zu erzählen, was ich so machte. Mir lag mehr an der Meinung der Leute, die Linux einsetzten. Ich hatte einen sozialen Zirkel geschaffen und mir den Respekt dieser Leute erworben. Das war zwar nicht mein Motiv, damals nicht und heute erst recht nicht. Aber das muss wohl das Wichtigste für mich gewesen sein. Deshalb habe ich so scharf auf Andrew Tanenbaum reagiert. "

Die Sonne beginnt im Pazifik zu versinken, und es ist Zeit, den Strand zu verlassen. Linus besteht darauf, dass ich auf der Rückfahrt fahre - um zu sehen, wie gut der Wagen reagiert - und dass wir die Route 9 nehmen, die längere, kurvenreiche Strecke ins Silicon Valley.

Linus sagt, dass er den Schlagabtausch mit dem Minix-Schöpfer schließlich per privater E-Mail führte, weil er zu hässlich geworden war, um öffentlich ausgetragen zu werden. Ein paar Monate lang war es ruhig. Dann schickte Tanenbaum Linus eine E-Mail, in der er ihn auf eine fünfzeilige Anzeige auf der Rückseite der Zeitschrift Byte hinwies, in der jemand eine kommerzielle Linux-Version bewarb.

„In der letzten E-Mail, die ich von Andrew bekam, fragte er mich, ob ich wirklich wollte, dass ein anderer meine Arbeit verkaufte. Ich schickte ihm eine E-Mail zurück, in der Ja stand, und habe seither nichts mehr von ihm gehört", sagt er.

Etwa ein Jahr später, als er in den Niederlanden war, um seinen ersten öffentlichen Vortrag zu halten, suchte Linus die Universität auf, in der Tanenbaum lehrte. Er wollte ihn bitten, Operating Systems: Design

and Implementation zu signieren, das Buch, das sein Leben so verändert hatte. Er wartete vor der Tür, aber Tanenbaum tauchte nicht auf. Der Professor war an diesem Tag nicht in der Stadt, und so sind sie sich nie begegnet.

XI.

Die Temperatur im Hotelzimmer war knapp über dem Gefrierpunkt, als ich in der Nacht vor meinem ersten Vortrag bibbernd im Bett lag. In Holland heizt man nicht wie in Finnland, und dieser zugige Raum hatte riesige, einfach verglaste Fenster, so als würde er normalerweise nur im Sommer bewohnt werden. Aber die Kälte war nicht das Einzige, was mich in der Nacht vom 4. November 1993 um den Schlaf brachte. Ich war über alle Maßen nervös.

Öffentliches Sprechen war mir immer schwer gefallen. In der Schule mussten wir Referate über Themen halten, die wir vorher recherchiert hatten - über Ratten oder so -, und ich war dazu einfach nie in der Lage gewesen. Ich stand vor der Klasse, brachte kein Wort heraus und fing an zu kichern. Und glauben Sie mir, ich bin niemand, der zum Kichern neigt. Ich fühlte mich sogar schon unwohl, wenn ich nur die Lösung einer Aufgabe an der Tafel vorführen sollte.

Aber nun war ich im niederländischen Ede, eine kurze Zugfahrt von Amsterdam entfernt, weil ich eingeladen worden war, die Eröffnungsansprache zum 10-jährigen Bestehen der niederländischen Unix Users Group zu halten. Ich wollte mir beweisen, dass ich das konnte. Ein Jahr davor war ich gebeten worden, vor einer ähnlichen Organisation in Spanien zu sprechen, aber ich hatte abgelehnt, weil meine Angst, öffentlich zu reden, größer war, als mein Wunsch zu reisen. Und damals machte es mir wirklich Spaß zu reisen. (Ich reise immer noch gern, aber es hat nicht mehr den Neuheitswert, den es für einen Jungen besaß, der kaum jemals aus Finnland herausgekommen war. Die einzigen Orte, in denen ich je gewesen war, waren Schweden, wo wir ein paar Zelturlaube verbrachten, und Moskau, wo wir meinen Vater besuchten, als ich ungefähr sechs Jahre alt war.)

Es ärgerte mich, dass ich die Chance vertan hatte, nach Spanien zu fahren, und deshalb nahm ich mir vor, die nächste Einladung zu einem Vortrag definitiv anzunehmen. Aber als ich in meinem Hotelbett lag und mich fragte, ob ich jemals meine Angst würde überwinden können, vor einer großen Gruppe von Menschen zu stehen, stiegen die Zweifel in mir hoch. Ich machte mir ernsthafte Sorgen, dass ich den Mund nicht aufbekommen oder, schlimmer noch, vor 400 Zuhörern zu kichern anfangen würde.

Ja, stimmt, ich gab ein Bild des Jammers ab.

Ich versuchte, mich mit den üblichen Floskeln zu beruhigen. Dass die Zuhörer möchten, dass man Erfolg hat, dass sie gar nicht da wären, wenn sie einen nicht leiden könnten, und dass ich mein Thema in- und auswendig kannte: die Gründe hinter den verschiedenen technischen Entscheidungen bei der Programmierung des Linux-Kernels, die Gründe, warum ich den Quellcode frei zugänglich machte. Aber all das konnte mich nicht davon überzeugen, dass der Vortrag ein Erfolg werden würde, und meine Schläfen pulsierten so unerbittlich wie die Lokomotive eines Güterzugs. Es schüttelte mich in meinem Bett. Die eisige Luft war daran am wenigsten schuld.

Der Vortrag? Na ja, das Publikum hatte Mitleid mit der verstörten Seele, die vor ihm stand, sich an ihren Powerpoint-Folien festhielt wie an einem Rettungsring (dem Himmel sei Dank für Microsoft) und dann zögernd die Zuhörerfragen beantwortete. Eigentlich war die Frage-und-Antwort-Runde der beste Teil. Nach meinem Vortrag -soweit man ihn als solchen bezeichnen konnte - kam Marshall Kirk McKusik, der damals eine wichtige Rolle bei BSD Unix spielte, zu mir herauf und sagte mir, er hätte meinen Vortrag interessant gefunden. Am liebsten wäre ich auf die Knie gefallen und hätte seine Füße geküsst. So dankbar war ich ihm für seine Geste. Es gibt in der Computerbranche nur wenige Leute, zu denen ich aufblicke, aber Kirk ist einer davon. Das liegt daran, dass er nach diesem ersten Vortrag so freundlich zu mir gewesen war.

Mein erster Vortrag war wie eine Schockbehandlung. Die nachfolgenden auch. Aber sie bauten mein Selbstvertrauen auf.

David fragt mich immer wieder, wie sich mein Status an der Uni änderte, als Linux immer größer wurde. Aber soweit ich mich erinnere, war es weder ein Thema bei den Professoren noch machten andere Studenten ihre Freunde auf mich aufmerksam. Nichts derglei-

chen. Die Leute, mit denen ich an der Uni zu tun hatte, wussten über Linux Bescheid, aber die meisten Hacker, die sich dafür engagierten, stammten nicht aus Finnland.

Im Herbst 1992 stellte mich der Fachbereich Informatik als Lehrassistenten für Kurse in schwedischer Sprache ein. (Das kam so: Sie brauchten schwedisch sprechende Lehrassistenten für die Computergrundkurse. Es gab nur zwei schwedisch sprechende Informatikstudenten, die bereits vor ein paar Jahren mit dem Studium begonnen hatten: Lars und Linus. Die Auswahl war nicht groß.) Zuerst hatte ich Angst, auch nur an die Tafel zu gehen und die Lösung einer Aufgabe vorzuführen, aber es dauerte nicht lange und ich konzentrierte mich auf den Stoff, statt zu fürchten, ich würde mich blamieren. Übrigens, drei Jahre später wurde ich zum Forschungsassistenten befördert und bekam mein Geld nicht mehr dafür, dass ich Studenten unterrichtete, sondern dafür, dass ich im Computerlabor arbeitete, was in der Hauptsache bedeutete, dass ich Linux weiterentwickelte. Das war der Beginn eines Trends: Andere zahlten mich dafür, dass ich Linux machte. Bei Transmeta ist es im Grunde genau so.

David: „Und wann begann dann ein Riesending daraus zu werden?“

Ich: „Es ist immer noch kein Riesending.“

Okay, ich nehme das zurück. Es wurde mehr daraus, als klar wurde, wie viele Leute Linux nicht nur als Spielzeug-Betriebssystem nutzten. Als sie es für mehr als nur zum Herumbasteln einzusetzen begannen, erkannte ich, dass ich für etwaige Probleme verantwortlich sein würde. Jedenfalls fing ich an, mich dafür verantwortlich zu fühlen. (Ich tue es immer noch.) Im Lauf des Jahres 1992 etablierte sich das System, das anfangs hauptsächlich ein Spielzeug gewesen war, im Leben seiner Anwender zu einer festen finanziellen und kommerziellen Größe.

Dieser Übergang ereignete sich im Frühjahr 1992, etwa ein Jahr, nachdem ich mit der Terminal-Emulation begonnen hatte, als die erste Version des X-Window-Systems unter Linux lief. Damit konnte das Betriebssystem eine grafische Benutzerschnittstelle unterstützen, und die Anwender erhielten die Möglichkeit, in mehreren Fenstern gleichzeitig zu arbeiten. Der Dank dafür gebührt dem X-Window-Projekt, das seinen Ursprung am MIT hatte. Das war ein Riesenschritt. Ich weiß noch, wie ich mit Lars etwa eineinhalb Jahre, bevor es wirklich so weit war, herumblödelte, wir würden eines Tages X ausführen und

dann ein Betriebssystem haben, das wirklich alles konnte. Ich hätte nie geglaubt, dass es so schnell geschehen würde. Einem Hacker namens Orest Zborowski gelang es, X nach Linux zu portieren.

Damit das X-Window-System funktioniert, braucht man den X-Server, der die ganze Grafik macht. Der Server redet mit den Clients, den Dingen, die sagen: „Ich will ein Fenster, das so und so groß sein soll.“ Die Kommunikation läuft über eine Schicht aus so genannten Sockets oder, förmlicher, Unix Domain Sockets. Sie werden für die interne Kommunikation in Unix, aber auch für die Kommunikation über das Internet verwendet. Deshalb schrieb Orest die erste Socket-Schicht für Linux, nur um X darauf zu portieren. Orests Socket-Schnittstelle wurde quasi angedockt und nicht mit dem anderen Code integriert. Das war eine der Situationen, in der ich einem Patch zustimmte, weil wir ihn brauchten, obwohl er ziemlich krude war.

Es dauerte eine Weile, bis ich mich an die Existenz der grafischen Benutzerschnittstelle gewöhnt hatte. Ich glaube, ich habe sie im ersten Jahr oder so nicht einmal täglich genutzt. Mittlerweile kann ich nicht mehr ohne sie leben. Wenn ich arbeite, sind immer Unmengen von Fenstern offen.

Orests Beitrag brachte uns nicht nur ein Fenstersystem ein, sondern öffnete auch eine Tür in die Zukunft. Die Domain Sockets wurden für die lokale Vernetzung benutzt, die es ermöglicht, X-Windows auszuführen. Mit den gleichen Sockets konnten wir Linux in die Lage versetzen, den Riesensprung zur externen Vernetzung zu tun und sich mit anderen Computern zu verknüpfen. Ohne Vernetzung war Linux nur für Leute nutzbar, die zu Hause saßen und sich über ein Modem irgendwo einwählten oder alles lokal erledigten. Mit großem Optimismus machten wir uns daran, eine Linux-Vernetzung auf diesen Original-Sockets zu entwickeln, obwohl sie überhaupt nicht für eine Vernetzung gedacht waren.

In meiner Überzeugung, dass uns das leicht gelingen würde, wagte ich einen kühnen Sprung in der Versionszählung. Eigentlich hatte ich geplant, im März 1992 die Version 0.13 freizugeben. Nun, wo die grafische Benutzerschnittstelle realisiert war, war ich sicher, dass wir unser Ziel, ein ausgewachsenes, zuverlässiges und obendrein netzwerkfähiges Betriebssystem freizugeben, zu etwa 95 Prozent erreicht hatten. Deshalb nannte ich die neue Freigabeversion 0.95.

Junge, Junge, das war vielleicht vorschnell. Und blauäugig.

Vernetzung ist ein ekelhaftes Geschäft, und wir mussten fast zwei Jahre investieren, um sie richtig hinzubekommen und eine Form zu finden, in der wir sie freigeben konnten. Wenn du Vernetzung hinzufügst, halst du dir mit einem Schlag eine Fülle neuer Probleme auf. Sicherheitsfragen müssen bedacht werden. Du weißt nicht, welche Leute da draußen sitzen und was sie im Schilde führen. Du musst wirklich darauf achten, dass die Leute deine Maschine nicht zum Absturz bringen, indem sie dir verseuchten Abfall zurückschicken. Du kannst nicht mehr kontrollieren, wer Kontakt zu deiner Maschine aufnimmt. Dazu kommt, dass viele Leute mit sehr unterschiedlichen Installationen arbeiten. Mit TCP/IP, dem Vernetzungsstandard, ist es schwierig, all die Timeouts richtig hinzubekommen. Die Sache schien sich ewig hinzuziehen. Ende 1993 hatten wir dann endlich eine fast brauchbare Netzwerkfähigkeit, obwohl manche Leute große Probleme hatten, sie zum Laufen zu bringen. Wir konnten keine Netzwerke ohne 8-Bit-Boundaries verwenden.

Wegen meiner allzu optimistischen Benennung der Version 0.95 war ich in einem Dilemma. Im Lauf der zwei Jahre, die wir brauchten, um Version 1.0 auf die Reihe zu bekommen, sahen wir uns zu ein paar verrückten Zahlenspielerereien gezwungen. Es gibt nicht viele Zahlen zwischen 95 und 100, und wir gaben ständig neue Versionen mit korrigierten Fehlern oder zusätzlichen Funktionen frei. Als wir bei Version 0.99 angekommen waren, fingen wir deshalb an, Patch-Levels durch zusätzliche Nummern zu kennzeichnen, und als Nächstes griffen wir auf das Alphabet zurück. Einmal hatten wir Version 0.99, Patch-Level 15 A. Darauf folgte Version 0.99, Patch-Level 15 B und so weiter. Wir brachten es bis auf Patch-Level 15 Z. Patch-Level 16 machten wir schließlich zur Version 1.0; die Netzwerkfähigkeit war endlich realisiert. Im März 1994 wurde sie mit großem Getöse im Haupthörsaal des Fachbereichs Informatik der Universität Helsinki freigegeben. Die Zeit davor war ziemlich chaotisch verlaufen.

Aber nichts konnte die Beliebtheit von Linux beeinträchtigen. Wir hatten unsere eigene Internet-Newsgroup, comp.os.linux, die aus den Überresten meines Schlagabtausches mit Andrew Tanenbaum erwachsen war. Sie zog scharenweise Leute an. Damals führte Internet Cabal, der enge Zirkel von Leuten, der mehr oder weniger für das Internet verantwortlich zeichnete, Monatsstatistiken darüber, wie viele Leser

jede Newsgroup anzog. Diese Statistiken waren nicht verlässlich, aber sie lieferten die besten Informationen, die man über die Popularität seiner Site beziehungsweise, in unserem Fall, über die Zahl der Linux-Interessenten bekommen konnte. Altsex war die unschlagbare Nummer eins. (Allerdings nicht für mich. Ich sah mir die Site zwar einoder zweimal an, um zu erfahren, was das Besondere daran war. Aber im Grunde war ich der typische sexuell minderbemittelte Nerd, der mehr darauf brannte, mit seinem Gleitkommaprozessor zu spielen, als sich über die jüngsten Neuigkeiten an der Sexfront auf dem Laufenden zu halten - neu entdeckte Positionen oder Berichte über Heavy Petting oder wie auch immer sonst die Themen heißen mochten, die die Leser von alt.sex bewegten.)

Mit der monatlichen Cabal-Statistik konnte ich die Popularität von comp.os.linux problemlos im Auge behalten. Und glauben Sie mir, ich behielt sie im Auge. (Manche mögen mich zwar für so eine Art Volkshelden halten, aber ich war nie der selbstlose, technikverliebte Unschuldengel ohne eigenes Ego, als den mich die halluzinierende Presse bevorzugt darstellt.) Im Herbst 1992 lag die Zahl der Zugriffe auf unsere Newsgroup in einer Größenordnung von zehntausenden von Leuten. So viele Leute verfolgten die Newsgroup, um sich auf dem Laufenden zu halten; sie setzten nicht alle Linux ein. Jeden Monat, wenn die Statistiken herauskamen, gab es einen Überblick über die 40 beliebtesten Newsgroups. Wenn deine Newsgroup es nicht unter die Top 40 geschafft hatte, konntest du die vollständige Übersicht über die Beliebtheit anderer Newsgroups bei einer speziellen Systemwartungs-Newsgroup abrufen. Normalerweise musste ich mir die Komplettübersicht holen.

Die Newsgroup kroch weiter auf den Charts nach oben. Irgendwann schaffte Linux es unter die Top 40, und ich war glücklich. Das war ziemlich cool. Wenn ich mich richtig erinnere, habe ich einen ziemlich schadenfrohen Artikel an comp.os.linux geschrieben, in dem ich die verschiedenen Betriebssystem-Newsgroups, einschließlich Minix, auflistete und in aller Deutlichkeit sagte: „Hey, schaut euch das an, wir sind beliebter als Windows.“ Allerdings waren die Windows-Anhänger, wie Sie vielleicht noch wissen, damals noch nicht im Internet zugange. Irgendwann 1993 schafften wir es unter die Top 5. An diesem Abend ging ich strotzend vor Selbstzufriedenheit zu Bett, bebend vor Aufregung, dass Linux fast so populär geworden war wie Sex.

Meine eigene kleine Ecke der Welt hatte demgegenüber ganz sicher nichts Vergleichbares zu bieten. Ich hatte damals kein richtiges Leben. Wirklich nicht. Zu- diesem Zeitpunkt hatte Peter Anvin, wie schon erwähnt, eine Online-Sammelaktion organisiert, die 3.000 Dollar an Spenden einbrachte, um mir zu helfen, meinen Computer abzubezahlen, was ich Ende 1993 auch tat. Und zu Weihnachten rüstete ich auf einen 486er DX266 um, den ich jahrelang hatte. Das war mein Leben: Ich aß. Ich schlief. Vielleicht ging ich zur Uni. Ich programmierte. Ich las eine Menge E-Mails. Mir war klar, dass manche meiner Freunde mehr Sex hatten, aber das war okay. Offen gesagt, die meisten meiner Freunde waren auch Loser.

MI

Der Vortrag in Ede hatte mich beinahe davon überzeugt, dass ich alles überleben konnte, selbst etwas so Furchteinflößendes wie vor einer Gruppe völlig fremder Menschen zu sprechen und im Mittelpunkt ihrer Aufmerksamkeit zu stehen. Auch in anderen Bereichen begann mein Selbstbewusstsein langsam zu wachsen. Ich war gezwungen, schnelle Entscheidungen über die Fehlerkorrekturen und Aktualisierungen von Linux zu treffen, und mit jeder Entscheidung fühlte ich mich in meiner Rolle als Oberhaupt einer wachsenden Gemeinde ein bisschen wohler. Die technischen Entscheidungen waren nie ein Problem für mich gewesen; mein Problem war es, einen Weg zu finden, einem Programmierer diplomatisch zu vermitteln, dass ich die Änderungsvorschläge eines anderen bevorzugte. Manchmal war es ganz einfach. Ich brauchte nur zu sagen: „Die Korrekturen von So-und-so funktionieren gut. Warum belassen wir es nicht einfach dabei?“

Ich habe nie einen Sinn darin gesehen, etwas anderes als den meiner Meinung nach besten technischen Lösungsvorschlag zu akzeptieren. Auf diese Weise brauchte ich keine Partei zu ergreifen, wenn mehrere Leute Verbesserungen zum gleichen Problem anboten. Und obwohl es mir damals noch nicht klar war, erwarb ich mir auf diese Weise auch das Vertrauen der Leute. Und Vertrauen zahlt sich aus. Wenn Leute dir vertrauen, hören sie auf deinen Rat.

Natürlich muss dieses Vertrauen auf irgendetwas gründen. In meinem Fall ist diese Vertrauensgrundlage vermutlich weniger der Linux-Kernel, den ich geschrieben habe, als vielmehr die Tatsache, dass ich ihn ins Internet gestellt und allen zugänglich gemacht habe, die daran mitarbeiten und die Funktionen und Details hinzufügen wollten, die ihnen gefielen, während die ultimativen Entscheidungen über den Kern des Betriebssystems mir vorbehalten blieben.

Genau so wenig wie ich je geplant habe, dass Linux ein Leben außerhalb meines eigenen Computers führen soll, habe ich geplant, sein geistiger Kopf zu sein. Das ergab sich einfach so. Irgendwann fing eine Kerngruppe von fünf Entwicklern an, sich um die meisten Aktivitäten in den entscheidenden Entwicklungsbereichen zu kümmern. Es war nahe liegend, dass sie als Filter fungierten und die Verantwortung für die Pflege dieser Bereiche übernahmen.

Ich lernte ziemlich früh, dass du Leute am besten und effektivsten führst, indem du ihnen keine Vorgaben machst, sondern sie einfach ihre Vorstellungen umsetzen lässt. Die besten Führungspersönlichkeiten wissen auch, wann sie falsch liegen, und sind in der Lage, einen Rückzieher zu machen. Und die besten Führungspersönlichkeiten versetzen andere in die Lage, Entscheidungen für sie zu treffen.

Oder anders ausgedrückt - ein Großteil des Erfolgs von Linux kann auf meine persönlichen Unzulänglichkeiten zurückgeführt werden: 1) Ich bin faul. Und 2) ich genieße es, Anerkennung für die Arbeit anderer einzuheimsen. Anderenfalls wäre das Linux-Entwicklungsmodell, wie die Leute es gerne nennen, immer noch auf tägliche E-Mails zwischen einer Hand voll Geeks beschränkt und nicht der Mittelpunkt eines fein gesponnenen Netzes aus hunderttausenden von Teilnehmern, das von Mailing-Listen, Entwicklerkonventionen und vielleicht 4.000 firmengesponserten, zeitgleich laufenden Projekten flankiert wird. An seiner Spitze steht ein Mensch, der dazu neigt und immer geneigt hat, nicht zu führen. Als eine Art Schiedsrichter schlichtet er Kontroversen über den Kern des Betriebssystems.

Etwas Besseres hätte mir nicht passieren können. Ich konnte mich der Dinge entledigen, die mich nicht besonders interessierten. Das fing mit der Benutzerebene an, den äußeren Teilen des Systems, mit denen Endanwender direkt interagieren, im Gegensatz zum internen Code, der tief im System verborgen liegt. Jemand bot sich an, die Pflege der UserEbene zu übernehmen. Nach und nach bildete sich ein organischer Prozess zur Pflege der verschiedenen Teilsysteme heraus. Die Leute wissen, wer aktiv und vertrauenswürdig ist, und alles Weitere ergibt sich von selbst. Keine Abstimmungen. Keine Befehle. Keine Nachzählungen.

Wenn zum Beispiel zwei Leute eine ähnliche Art von Softwaretreiber geschrieben haben, akzeptiere ich manchmal die Arbeit beider und warte ab, welche Lösung sich am Ende durchsetzen wird. Anwender neigen dazu, dem einen oder dem anderen den Vorzug zu geben. Oder

ich fordere die Konkurrenten auf, die Sache unter sich auszumachen, und wenn ich Glück habe, entwickeln sich ihre Ansätze in unterschiedliche Richtungen und decken letztlich unterschiedliche Aufgaben ab.

Viele Leute sind erstaunt, dass das Open-Source-Modell tatsächlich funktioniert.

Ich nehme an, dazu muss man die Mentalität der Hacker im freien Softwareuniversum verstehen. (Übrigens, normalerweise versuche ich, den Begriff „Hacker“ zu vermeiden. Ich habe mich zwar früher in Gesprächen mit technisch orientierten Leuten durchaus selbst als Hacker bezeichnet. Aber inzwischen hat der Begriff eine andere Bedeutung angenommen: minderjährige Kids, die nichts Besseres zu tun haben, als herumzusitzen und elektronisch in Rechenzentren einzubrechen, während sie eigentlich ehrenamtlich in der Gemeindebücherei arbeiten oder wenigstens ihre Freundin befummeln sollten.)

Die Hacker, äh, Programmierer, die an Linux oder anderen Open-Source-Projekten arbeiten, verzichten darauf, zu schlafen, sich auf dem Stairmaster fit zu halten, bei den Little-League-Spielen ihrer Kinder dabei zu sein und manchmal sogar darauf, Sex zu haben, weil sie leidenschaftlich gern programmieren. Und sie finden es genial, Teil einer globalen Gemeinschaftsanstrengung zu sein - Linux ist das größte Gemeinschaftsprojekt der Welt -, die sich der Aufgabe verschrieben hat, die beste und schönste Technologie zu schaffen und jedem zugänglich zu machen, der sie haben möchte. Das ist das ganze Geheimnis. Und das macht Spaß.

Okay, mit all dieser schamlosen Selbst-PR klinge ich allmählich wie eine Pressemitteilung. Open-Source-Hacker sind nicht das Hightechfendant zu Mutter Teresa. Ihre Beiträge werden namentlich gewürdigt - in Form einer Credit List oder eines History File, die mit jedem Projekt verbunden sind. Die produktivsten Mitwirkenden ziehen die Aufmerksamkeit von Arbeitgebern auf sich, die den Code in der Hoffnung durchforsten, Spitzenprogrammierer zu finden und anzuheuern. Darüber hinaus sind die meisten Hacker auch durch die Anerkennung motiviert, die sie in den Augen ihrer Peers durch solide Beiträge gewinnen. Dieser Motivationsfaktor ist nicht zu unterschätzen. Jeder will seine Peers beeindrucken, seinen Ruhm mehren, seinen sozialen Status erhöhen. Die Open-Source-Entwicklung eröffnet Programmierern eine solche Chance.

Selbstredend verbrachte ich einen Großteil des Jahres 1993 wie ich einen Großteil der Jahre 1992, 1991 usw. verbracht hatte: über einen Computer gebeugt. Das sollte sich ändern.

Den akademischen Fußstapfen meines Großvaters folgend war ich Lehrassistent an der Universität Helsinki und im Herbstsemester für den Kurs „Einführung in die Informatik“ in schwedischer Sprache verantwortlich. So lernte ich Tove kennen. Sie hatte einen noch größeren Einfluss auf mein Leben als das Buch *Operating Systems: Design and Implementation*. Aber ich will Sie nicht mit all zu vielen dieser Details langweilen.

Tove gehörte zu den fünfzehn Teilnehmern des Kurses. Sie hatte bereits einen Abschluss für Vorschulerziehung. Sie wollte daneben noch Informatik studieren, kam aber nicht so schnell voran wie die anderen Kursteilnehmer. Aber letztlich holte sie die anderen ein.

Weil der Kurs so elementar und das Internet im Herbst 1993 weit von seiner heutigen Popularität entfernt war, sollten die Teilnehmer mir eines Tages als Hausaufgabe eine E-Mail senden. Heute klingt es absurd, aber ich sagte: „Als Hausaufgabe schickt ihr mir eine E-Mail.“

Die E-Mails der anderen Studenten enthielten simple Testbotschaften oder nichts sagende Bemerkungen über den Kurs.

Tove lud mich ein, mit ihr auszugehen.

Ich heiratete die erste Frau, die mich elektronisch aufriss.

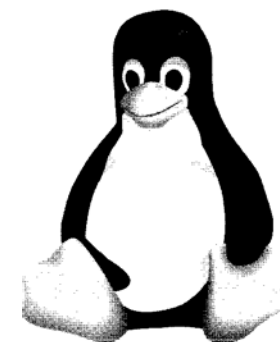
Unsere erste Verabredung ging nie zu Ende. Tove war Kindergärtnerin und sechsfache finnische Karatemeisterin, das Produkt einer funktionalen Familie, obwohl ich vermutlich jede Familie so beschreiben würde, die nicht so skurril ist wie meine. Sie hatte eine Menge Freunde. Vom ersten Moment an hatte ich das Gefühl, sie sei die richtige Frau für mich. (Ich erspare Ihnen weitere Ausführungen dazu.) Nach wenigen Monaten zogen Randi, der Kater, und ich in ihre winzige Wohnung ein.

In den ersten zwei Wochen machte ich mir nicht einmal die Mühe, meinen Computer mitzubringen. Wenn ich meinen Wehrdienst nicht mitzähle, waren diese beiden Wochen die längste Zeit in meinem Leben, die ich fern von einem Computer verbrachte, seit ich als Elfjähriger auf dem Schoß meines Großvaters gesessen hatte. Ohne näher darauf einzugehen, ist dieser Rekord bis heute ungebrochen. Irgendwie gelang es mir. (Wieder spielen die Details keine Rolle.) Meine

Mutter pflegte die seltenen Male, die ich sie in jener Zeit sah, irgendeinen Unsinn über „einen Triumph von Mutter Natur“ zu murmeln. Meine Schwester und mein Vater waren, glaube ich, einfach sprachlos.

Es dauerte nicht lange, und Tove legte sich eine Katze zu, die Randi Gesellschaft leistete. Nach und nach verfielen wir in eine angenehme Routine, die Abende allein oder mit Freunden zu verbringen und dann am nächsten Morgen um fünf aufzuwachen, so dass sie zur Arbeit und ich früh genug zur Uni gehen konnte, um allein und ungestört meine Linux-Mails zu lesen.

Ballkönig



Die Geburt von Version 1.0 bedeutete etwas Neues für Linux: die Notwendigkeit, für PR zu sorgen. Ich wäre ebenso glücklich gewesen, die neue Version der Welt mehr oder weniger auf die gleiche Weise vorzustellen wie die Vorgängerversionen. Ich hätte in der Newsgroup etwas in der Art geschrieben wie: „Version 1.0 ist draußen. Tut damit, was ihr wollt.“ (Okay, ich hätte es etwas anders formuliert.)

Eine Menge anderer Leute sahen ein sehr viel größeres Ereignis darin. Sie waren aus Marketing-Gründen an Version 1.0 interessiert. Mittlerweile gab es all die aufstrebenden kommerziellen Firmen, die angefangen hatten, Linux zu verkaufen. Für sie war Version 1.0 aus psychologischen, nicht aus technischen Gründen wichtig. Ich konnte es ihnen nicht verdenken. Tatsache ist, es sieht einfach schlecht aus, wenn du versuchst, Version 0.96 eines Betriebssystems zu verkaufen.

Ich wollte die Version draußen haben, weil sie einen Meilenstein für mich darstellte, und weil ich nach erfolgter Freigabe eine Zeit lang keine Fehler mehr zu beheben brauchte und mich wieder der Entwicklung zuwenden konnte. Die Firmen und die Linux-Gemeinde wollten die Öffentlichkeit im großen Stil mit der Version bekannt machen.

Wir brauchten eine Public-Relations-Strategie. Ich war nicht bereit, mich dafür persönlich zu engagieren. Ich hatte kein Interesse, Pressemitteilungen zu versenden oder Statements abzugeben. Andere hielten das für nötig und boten deshalb an, sich darum zu kümmern. Es war so ähnlich wie bei Linux selbst - irgendwie funktionierte es.

Lars gehörte zu den treibenden Kräften im Hintergrund, die die erste offizielle Freigabe zu einem echten Event machten. Er und ein paar andere hielten die Universität für den angemessensten Ort für die Ankündigung. Das war sinnvoll. Mein Zimmer war zu klein dafür.



Und es hätte einen unguten Präzedenzfall geschaffen, wenn sich ein kommerzielles Unternehmen als Gastgeber für die Ankündigung zur Verfügung gestellt hätte. Deshalb erklärte Lars sich bereit, die Veranstaltung mit der Universität zu koordinieren. Der Fachbereich Informatik der Universität Helsinki war so klein, dass er die Angelegenheit direkt mit dem Dekan regeln konnte.

Die Universität Helsinki war sehr gerne bereit, den Haupthörsaal des Fachbereichs Informatik für die Einführung der Linux-Version 1.0 zur Verfügung zu stellen. Und warum auch nicht? Wie oft schafft es eine Universität schon, in die Fernsehnachrichten zu kommen?

Und ich willigte ein, einen Vortrag zu halten. Wobei dieser Vortrag in keiner Weise die Schrecken meines Ede-Traumas in sich barg. Okay, bei näherer Betrachtung war manches eigentlich *noch* schwieriger.

Zum Beispiel, dass mein Vater im Publikum saß. Und die Tatsache, dass der Vortrag im finnischen Fernsehen gesendet wurde. Es war das erste Mal, dass ich die Gelegenheit hatte, mich im Fernsehen zu sehen. Meine Eltern saßen beide im Publikum (wenn auch, da bin ich mir ziemlich sicher, nicht nebeneinander). Tove war auch da. Es war das erste Mal, dass mein Vater Tove traf, deshalb bedeutete das Ereignis für mich mehr als nur die Ankündigung von Version 1.0. Da ich mit den letzten Vortragsvorbereitungen beschäftigt war, zum Beispiel der Überprüfung meiner Folien, war ich bei dieser ersten Begegnung nicht dabei. Anscheinend fand sie auf dem Weg in den Hörsaal statt. Vielleicht habe ich sie aus den Augenwinkeln mitbekommen.

In diesem Vortrag wie in praktisch jedem anderen in den darauffolgenden Jahren sprach ich weniger über die Technik als über Open-Source. Es lief gut. Die Veranstaltung änderte ein paar Meinungen über Linux im Fachbereich Informatik. Vorher hatte der Fachbereich mit nachsichtigem Stolz und freundlicher Ermutigung auf Linux reagiert. Nach der Ankündigung begannen die Leute im Fachbereich Linux ernster zu nehmen. Schließlich hatten sie es in den Nachrichten gesehen. Im Lauf der Jahre wurde gelegentlich gemunkelt, die Universität versuche, sich das Verdienst für Linux auf ihre Fahnen zu schreiben. Das war nicht der Fall. Der Fachbereich war immer sehr entgegenkommend. Sie gaben mir einen Job, so dass ich auf ihre Kosten an Linux arbeiten konnte. Wohlgemerkt, das war in den Anfangstagen, wo noch niemand sagen konnte: „Lass uns auf den Zug aufspringen, denn

eines Tages wird das Zeug berühmt sein.“ Aber gleichzeitig kam es der Uni natürlich sehr gelegen, eine wichtige Rolle bei der Ankündigung zu spielen. Das brachte ihr eine großartige PR ein. Ich weiß, dass heute deutlich mehr schwedisch sprechende Studenten am Fachbereich Informatik studieren, der vorher immer im Schatten der technischen Universität gestanden hatte.

Erfolgsneid gilt als finnische Charaktereigenschaft. Und als Linux der Welt ein Begriff zu werden begann, wurde ich häufig gefragt, ob die Leute an der Uni neidisch auf mich waren. Das Gegenteil war der Fall. Sie unterstützten mich, wo sie nur konnten. Zum Beispiel trennten sie sich sehr früh von den X-Terminals und installierten stattdessen Linux-PCs.

Die Ankündigung katapultierte Linux aus dem finnischen Dunstkreis hinaus und sorgte auch anderswo für öffentliches Interesse. Viele der frühen Schlagzeilen kamen dadurch zustande, dass ein Journalist über Linux stolperte und es aufregend fand. Vom geschäftlichen Standpunkt aus gesehen stellte Version 1.0 nie eine besondere Herausforderung für die Big Players dar. Linux begann, in den Markt einzudringen, den bisher Minix und Coherent innehatten. Aber außerhalb dieser Gemeinde war die Aufmerksamkeit gering, und das war okay. Der Erfolg war ohnehin viel größer, als ich anfangs erwartet hatte.

Dessen ungeachtet begannen Journalisten, meistens von Fachzeitschriften, an meine Tür zu klopfen - im wahrsten Sinne des Wortes. Es machte Tove nicht glücklich, an einem Samstagmorgen aufzuwachen, um einen japanischen Reporter vorzufinden, der mit Geschenken beladen war - meistens Uhren, sie hatten wahrscheinlich irgendwo gehört, dass ich eine Schwäche für Uhren habe -, und ein Interview mit mir führen wollte. Es machte Tove noch weniger glücklich, wenn ich den Störenfried herein bat. (Diese Gewohnheit sollte ich jahrelang beibehalten, bis wir unser neues Haus zur journalistenfreien Zone erklärten. In meinen rücksichtslosesten Momenten vergaß ich nicht nur, Tove zu sagen, dass ich einen Journalisten nach Hause eingeladen hatte, der ein Interview mit mir machen wollte, sondern verpennte auch noch den Termin. Wenn der Reporter dann auftauchte, musste Tove ihn unterhalten, bis ich endlich nach Hause kam.) Dann entstanden die ersten Fan-Websites, zum Beispiel eine in Frankreich, die hauptsächlich aus einer häufig aktualisierten Galerie peinlicher Fotos von mir bestand. Eines davon zeigt mich bei einem Spektrum-Treffen:

Ich habe kein Hemd an, halte ein Bier in der Hand und sehe ziemlich trinkfest aus.

O Mann.

Aber nicht nur Journalisten oder Linux-Hackertypen interessierten sich für mich. Plötzlich wollten Leute mit großen Spesenkonten mit mir über ihre Technologie sprechen. Unix galt seit langem als äußerst viel versprechendes Betriebssystem, vor allem wegen seinen Leistungseigenschaften und Multitasking-Fähigkeiten. Deshalb begannen Firmen, die sich für Unix interessierten, Linux im Auge zu behalten. Zu ihnen gehörte auch das Netzwerkunternehmen Novell, das ein hirnrissiges Projekt auf Linux-Basis begonnen hatte: Sie entwickelten einen Unix-Desktop namens Looking Glass. Er sah gut aus, hatte aber einen entscheidenden Fehler: Er entsprach nicht dem damals üblichen Standard, dem Common Desktop Environment.

Im August 1994 luden sie mich nach Orem, Utah, ein, um mit ihnen über ihren Desktop zu sprechen. Novell bot mir damit die große Chance, Amerika zu sehen, und deshalb sagte ich, ich würde kommen, wenn sie mir zusätzlich einen Besuch in einer anderen amerikanischen Stadt finanzieren würden. Selbst ich als weltfremder Finne ahnte vage, dass Orem - oder auch Salt Lake City - nicht unbedingt repräsentativ für den Rest des Landes sein würde. Die Novell-Leute schlugen Washington vor, aber dorthin wollte ich nicht. Ich nahm an, es wäre eine Hauptstadt wie jede andere. Sie schlugen New York vor, aber ich wollte lieber Kalifornien sehen.

In der Novell-Zentrale war kaum erkennbar, wie ernst es den Leuten mit dem Projekt war. (Am Ende stellte sich heraus, dass es ihnen nicht besonders ernst damit war; sie ließen das Projekt platzen, und die neun Beteiligten gründeten Caldera.) Aber ich bekam einen ersten Eindruck von den Vereinigten Staaten, wo ich, davon war ich überzeugt, irgendwann in meinem Leben leben würde. Ganz unabhängig von dem Engagement, das Novell für Linux zeigte, schienen mir die USA das Zentrum des wachsenden technologischen Universums zu sein.

Mein Besuch in den USA war ein Kulturschock. Als Erstes fiel mir auf, wie neu alles war, verglichen mit Europa. Die mormonische Kirche hatte ein paar Jahre vor meinem Besuch den 150. Jahrestag ihres Bestehens gefeiert, und anlässlich dieses Ereignisses die Fassade ihres Haupttempels reinigen lassen. Sie leuchtete in strahlendem Weiß. Aus

Europa kommend, wo alle Kirchen alt sind und der Zahn der Zeit seine Spuren hinterlassen hat, konnte ich beim Anblick des weißen Tempels nur eines denken: Disneyland. Der Tempel ähnelte eher einem Märchenschloss als einer Kirche. Und in Orem beging ich den Fehler, mir die Sauna des Hotels anzusehen. Es war eine dieser Porta-PottySaunas aus Plastik, und drinnen war es kaum wärmer als draußen. Danach wusste ich, dass die Amerikaner keine Saunas bauen können, und hatte ein wenig Heimweh.

Und ich begann, den Bogen rauszukriegen. Genauso wie Besucher in Finnland schnell lernen, keine x-beliebigen Fremden in Bars anzusprechen, lernte ich, dass es in Utah - und, wie ich später merkte, auch im Rest der Vereinigten Staaten - unmöglich ist, rational über Themen wie Abtreibung oder Gewehre zu diskutieren. Es besteht eine Chance von

etwa fünfzig Prozent, das dein Gegenüber diese Themen sehr emotional sieht, und es ist so einfach, in einen Riesenstreit über etwas zu geraten, über das man nicht streiten sollte. Europäer sind weniger von diesen Themen besessen. In den USA beharren die Leute so erbittert auf ihren Standpunkten, weil sie sich die Gegenmeinung so oft anhören mussten. Es gibt in Finnland wahrscheinlich mehr Gewehre pro Kopf als in vielen anderen Ländern, aber sie werden hauptsächlich zur Jagd genutzt. Ansonsten sind sie bei uns kein Thema.

Und noch etwas lernte ich in meinen ersten Tagen auf U.S.-amerikanischem Boden sehr schnell: Root-Beer schmeckt scheußlich.

Von Utah aus flog ich nach San Francisco weiter und dort gefiel es mir unglaublich gut. Ich lief so lange in der Stadt herum, dass ich einen schweren Sonnenbrand bekam und einen ganzen Tag in geschlossenen Räumen verbringen musste.

Ich weiß noch, wie ich über die Golden Gate Bridge ging, zu den Marin Headlands hochblickte und mir begeistert die Möglichkeit aus-

''' Junge, das Zeug ist gewöhnungsbedürftig. Ich nehme an, es geht auf die Puritaner zurück, die kein Bier trinken durften, weil darin Alkohol enthalten ist. Deshalb brauten sie etwas anderes zusammen, das keinen Alkohol, dafür aber Wurzeln enthielt (engl. roots), und nannten es Root-Beer, in der Hoffnung, die Leute würden darauf hereinfallen und es für gut befinden. Nachdem zehn Generationen von Leuten darauf hereingefallen waren und es für gut befunden hatten, hatten sie ihr Ziel erreicht: Heute mögen die Amerikaner Root-Beer, weil sich ihre Gene in zehn Generationen daran angepasst haben.

malte, dort wandern zu können, gleich nachdem ich die Brücke überquert hatte. Aber als ich es auf die Marin-Seite geschafft hatte, hatte ich jedes Interesse daran verloren, noch weiter zu gehen. Ich hätte es mir nie träumen lassen, dass ich sechs Jahre später, fast auf den Tag genau, auf dem Kamm dieser windigen Headlands sitzen, über den Pazifik, die San Francisco Bay, die Brücke, den Nebel und San Francisco selbst hinausblicken und meine Gedanken auf Davids Kassettenrekorder sprechen würde.

Es dauerte nur ein Jahr, bis ich wieder in die USA flog. Ich kam zurück, um einen Vortrag bei der DECUS zu halten, der Usergroup von Digital in New Orleans. Dieses Mal kam Tove mit. Bei dem Treffen waren nur vierzig Leute anwesend, deshalb hielt sich meine Qual in Grenzen. Das Beste daran war, dass ich Maddog, dessen bürgerlicher Name Jon Hall ist, kennen lernte. Er arbeitete im technischen Marketing für Digital Unix und war ein ausgefuchster Unix-Veteran. Meine Einladung zu dem Vortrag verdankte ich ihm. Maddog, der für seinen bis auf die Brust reichenden Bart und seinen absurden Sinn für Humor (ganz zu schweigen von seiner Neigung zu schnarchen) bekannt ist, leitet Linux International, eine Organisation, die sich der Unterstützung von Linux und Linux-Anwendern verschrieben hat. Außerdem ist er der Pate meiner Tochter Patricia.

Ein weiteres Vermächtnis dieses Treffens in New Orleans: Maddog arrangierte, dass Digital mir einen Alpha zur Verfügung stellte. Auf diese Weise wurde Linux auf etwas anderes als einen PC portiert. Zwar hatten Leute schon vorher Linux auf andere Architekturen portiert. Es gab eine Portierung auf die 68K-Familie, jene Motorola-Prozessoren, die in den Ataris und Amigas eingebaut waren. Aber in diesen Fällen lief Linux nicht gleichzeitig auf beiden Plattformen. Stattdessen mussten Teile, die auf der neuen Plattform nicht funktionierten, rausgeworfen und neue Teile dazugeschrieben werden. Deshalb war die erste echte Linux-Portierung die auf den Alpha. Im Prinzip lief sowohl auf dem PC als auch dem Alpha der gleiche Quellcode. Zu diesem Zweck wird eine Abstraktionsebene eingefügt, so dass der gleiche Code auf zwei verschiedene Arten kompiliert wird, um auf zwei verschiedenen Architekturen laufen zu können. Der Code bleibt der Gleiche, kann aber auf verschiedenen Architekturen genutzt werden.

Als wir Version 1.2 im März 1995 freigaben, war der Kernel auf 250000 Zeilen Code angewachsen, die neue Zeitschrift *Linux Journal* konnte auf 10000 Leser verweisen, und Linux war auf Intel-, Digital- und Sun-SPARC-Prozessoren lauffähig. Das war ein Riesenschritt.

II.

Es ist 1995, eine Fülle kommerzieller Versionen sind dabei, Fuß zu fassen, und Linux-Firmen erfreuen sich einer wachsenden Anhängerschaft. Die Universität hat mich vom Lehrassistenten zum Forschungsassistenten befördert, und das bedeutet mehr Geld und weniger Lehrverpflichtungen. Langsam - sehr langsam - erwerbe ich die Scheine, die ich noch für meine Master-Arbeit brauche, bei der es um die Portierung von Linux auf verschiedene Architekturen geht. Tove hat mir Squash beigebracht, und wir machen einmal in der Woche zusammen ein Spiel; wir sind einander ziemlich ebenbürtig.

In diese glückselige Beschaulichkeit platzt ein Problem. Es stellt sich heraus, dass ein opportunistischer Typ aus Boston das Wort Linux als Warenzeichen eintragen ließ. Nicht nur das, er schickt auch E-Mails an das *Linux Journal* und ein paar Linux-Firmen mit der Forderung, ihm fünf Prozent ihrer Einkünfte als „Dankeschön“ für das Warenzeichen zu überweisen.

Als ich davon hörte, beschlich mich ein Déjà-vu-Gefühl. Der Name des Burschen kam mir bekannt vor. Ich sah meine E-Mail-Ablage durch. Vor etwa eineinhalb Jahren hatte er mir unaufgefordert eine E-Mail gesandt, in der er wissen wollte, ob ich an Gott glaubte, und behauptete, er hätte eine grandiose Geschäftsidee für mich. Das war in einer Zeit, in der der Wust unaufgeforderter E-Mails noch nicht zur globalen Obsession geworden war. In jenen unschuldigen Jahren kam es niemandem in den Sinn, das Internet mit Werden-Sie-schnellreich-Angeboten zuzumüllen. Nein, ich machte mir nicht die Mühe, auf die E-Mail des Typs zu reagieren, aber weil sie damals so ungewöhnlich war, hatte ich sie gespeichert.

Somit waren wir mit einer kleinen Krise konfrontiert. Wir waren Hacker. Niemand hatte daran gedacht, das Markenregister einzusehen.

Der Typ war kein professioneller Warenzeichen-Besetzer. Er hatte es wohl nur dieses eine Mal versucht. Warenzeichen sind in verschiedene Klassen eingeteilt. Er trug seines in der Klasse Computer ein. Man muss bei der Beantragung eines Warenzeichens Beweise vorlegen, und so legte er der Registerstelle eine Diskette vor und behauptete, es befände sich ein Programm namens Linux darauf.

Wir gerieten in Panik. Jedem in der Linux-Gemeinde war klar, dass wir um das Warenzeichen kämpfen würden. Das Problem dabei war, dass wir nicht die Organisation hatten, uns wirksam zu wehren. Wir hatten nicht einmal genug Geld, uns einen Anwalt zu nehmen. Keine der Linux-Firmen wollte gern die geforderte Summe in Höhe von 15 000 Dollar auf den Tisch legen. (Heute geben sie locker so viel Geld im Monat für Mountain-Dew* aus.) Aber damals war das für ein einzelnes Unternehmen eine erhebliche Summe Geld. Deshalb beschlossen das *Linux Journal* und ein paar andere Firmen, Geld in die Linux-International-Organisation zu pumpen, die den Rechtsstreit um das Warenzeichen führen sollte. Linux International war in Australien von einem gewissen Patrick D'Cruze gegründet worden, übersiedelte aber 1994 in die USA, um an der weltweiten Förderung von Linux mitzuwirken. Das Jahr des Warenzeichenstreits war das Jahr, in dem Maddog die Leitung von Linux International übernahm. Er genoss und genießt allgemeines Vertrauen.

Ich war in Finnland, versuchte Tove beim Squash oder Avuton beim Billard zu schlagen und hatte nicht das geringste Interesse daran, in die Sache hineingezogen zu werden. Ich wollte einfach, dass der ganze Albtraum sich in Luft auflöste. Nach meiner damaligen Vorstellung hätten wir versuchen sollen, das Warenzeichen loszuwerden und darauf hinzuwirken, es für ungültig erklären zu lassen, weil es bereits vorher in der Branche kommerziell genutzt worden war. Tatsächlich hatten wir genügend Belege, um zu beweisen, dass Linux schon früher verwendet worden war. Allerdings überzeugte uns der Anwalt, dass sich der Aufwand nicht lohnte, ja, dass wir nicht einmal versuchen

Anmerkung der Übersetzerin: Ein grünelber, stark koffeinhaltiger Softdrink.

sollten, Linux als Public Domain statt als Warenzeichen eintragen zu lassen. Die einzige Möglichkeit, dass Linux als Public Domain anerkannt würde, bestünde darin, dass es zum Allgemeingut wurde. Aber Linux war damals eigentlich kein Allgemeingut. Die Warenzeichenstelle des Patentamts würde es wahrscheinlich noch nicht einmal heute als Allgemeingut betrachten. Wir könnten die Schlacht verlieren, sagte er. Und wenn wir alternativ dazu das alte Warenzeichen für ungültig erklärten, könnte jemand kommen und es neu eintragen lassen.

Der Anwalt schlug deshalb vor, das Warenzeichen auf jemand anderen übertragen zu lassen. Ich war für Linux International. Aber dagegen gab es eine Menge Einwände. Die Leute hatten Angst, bei Linux International könnten die kommerziellen Interessen obsiegen. (Heute wissen wir: Das war nicht der Fall.) Es bestanden große Bedenken, wer eines Tages bei einem Rücktritt von Maddog dessen Nachfolger werden würde.

Deshalb richteten sich alle Augen auf mich. Linux International war neu und unerprobt. Der Anwalt gab zu bedenken, dass die Beweislage einfacher sein würde, wenn das Warenzeichen auf mich übertragen würde, weil ich den Begriff als Erster verwendet hatte. Diese Strategie setzten wir ein. Wir einigten uns außergerichtlich, weil uns das am einfachsten und billigsten zu sein schien. Wie bei den meisten Vergleichen dürfen die Details nicht offen gelegt werden. Nicht, dass ich sie überhaupt kennen würde. Ich war glücklich und zufrieden, möglichst wenig davon mitzubekommen.

Als ich später die erste E-Mail nachlas, die der Typ mir geschickt hatte, erkannte ich, dass darin von Warenzeichen keine Rede war. Es war offensichtlich, dass er nur mit mir reden wollte. Vielleicht versuchte er, Kontakt mit mir aufzunehmen, um mich dazu zu bringen, ihn abzufinden. Oder vielleicht hätte er mir, wenn ich mich als Rechtgläubiger und Seelenverwandter im Glauben erwiesen hätte, das Warenzeichen einfach *geschenkt*. Ich weiß es nicht.

Ich akzeptiere, dass manche Leute kein Gefühl für Moral haben. Wirklich irritierend fand ich damals jedoch die Tatsache, dass das ganze Warenzeichensystem mir, der ich nichts Unrechtes getan hatte, die Last auferlegte, hinauszugehen und den Typ zu bekämpfen.

Als Ergebnis dieses schmutzigen Gefechts bin ich heute Inhaber des Linux-Warenzeichens. Das heißt, wenn Unternehmen wie VA Linux an die Börse gehen wollen, muss ihr Verkaufsprospekt auf die Tatsa-

che hinweisen, dass das Unternehmen noch nicht einmal für die Hälfte seines Namens das Warenzeichen besitzt. (In diesem Fall war das Unternehmen an den gerichtlichen Schritten beteiligt, die mir die Nutzung des Begriffes Linux sicherten.) Aber an Vorkommnisse dieser Art habe ich mich gewöhnt.

Die Episode um das Warenzeichen war für Linux nur ein unerwarteter Wachstumsschmerz. Und eine Ablenkung. Aber kaum war sie vorbei, tauchte schon eine neue auf: Ein Ingenieur am Intel-Forschungslabor in Portland, Oregon, teilte mir mit, sein Unternehmen setze Linux bei der Erforschung neuer Architekturen ein. Er fragte mich, ob ich für ein sechsmonatiges Praktikum herüberkommen wollte.

Tove und ich hatten vage darüber gesprochen, uns möglicherweise in den USA niederzulassen. Sie wusste, wie sehr ich meine ersten Amerikabesuche trotz des Root-Beers genossen hatte. Wir waren uns einig, dass die Chancen - vom Klima ganz zu schweigen - in den USA besser sein würden. (Übrigens bin ich völlig überzeugt davon, dass das amerikanische System der Mitarbeitermotivation weitaus realistischer und produktiver ist als das europäische Modell. Wenn in Finnland ein Mitarbeiter viel mehr leistet als seine Kollegen, gibst du ihm etwas mehr Geld und sorgst dafür, dass es möglichst niemand erfährt. In den USA gibst du ihm viel mehr Geld - und es funktioniert.) Das Praktikum erschien uns als geniale Möglichkeit, das Wasser oder vielmehr - angesichts der geografischen Lage von Portland im Nordwesten der USA - das Regenwasser zu testen, und wir beschlossen, dass ich die Chance nutzen sollte. Aber meine Gefühle waren zwiespältig. Ich hatte ein ungutes Gefühl, ohne Master-Abschluss von der Uni wegzugehen. Vielleicht war die Erinnerung an meinen Großvater, den Professor, daran schuld, auf jeden Fall widerstrebte mir der Gedanke, ein Studienabbrecher zu sein. Am Ende spielten meine Gefühle jedoch keine Rolle. Der Manager des Ingenieurs kam zu dem Schluss, dass es mir schwer fallen dürfte, die für das Praktikum erforderliche sechsmonatige Arbeitserlaubnis des U.S. Immigration and Naturalization Service zu bekommen.

Also blieben wir in Helsinki. Als Finnlands legendäre Nachtschwärmer auf die Ankunft des Neujahrstags 1996 anstießen, schob ich mich zentimeterweise der Ziellinie des Master-Abschlusses entgegen. Ich brauchte nur noch einen kleinen Kurs abzuschließen, um alle Scheine beisammen zu haben. Und ich musste meine Abschlussarbeit schrei-

ben. Ironischerweise würde ich dann zum ersten Mal eine akademische Anerkennung für meine Arbeit an Linux bekommen, die während der Jahre an der Uni fast mein ganzes Leben verschlungen hatte.

Das Jahr 1996 rüttelte mich wach. Im egalitären Finnland bekommst du eine staatlich vorgeschriebene Dienstalterszulage, wenn du einen Job drei Jahre lang ausgeübt hast. Als ich zum ersten Mal die Gehaltsabrechnung mit dem neuen Gehalt sah, durchzuckte mich ein Schock: Ich hatte lange genug an der Universität gearbeitet, um eine Dienstalterszulage zu erhalten. Würde ich mein gesamtes Berufsleben dort verbringen? War ich dazu bestimmt, wie mein Großvater zu enden? Sie erinnern sich, wie ich ihn weiter vorn in diesem Buch beschrieben habe: kahlköpfig, übergewichtig und ohne charakteristischen Geruch? Ich fing an, mich ziemlich regelmäßig im Spiegel zu betrachten. Mein Haaransatz hatte sich um ein paar Millimeter nach hinten verschoben. Auf meinem einstmals flachen Bauch schienen sich zusätzliche Pfunde angesiedelt zu haben. Ich war 26 und zum ersten Mal in meinem Leben fühlte ich mich alt. Ich hing seit fast sieben Jahren an der Uni herum. Ich wusste, wenn ich mich zusammenriss, würde ich meinen Abschluss ziemlich schnell schaffen können.

*Meine zehnjährige Tochter Kaley denkt, dass du der größte aller Superstars bist, wenn jemand dir einen Pinguin schenkt. Wir sitzen an einem klaren Sierra-Abend am Lagerfeuer, als Linus erzählt, eine Linux-Usergroup in Bristol in England habe ihm einen Pinguin gekauft. Für Kaley ist es unvorstellbar, dass er sich nicht die Mühe machte, den Vogel zu besuchen. Dann klärt er uns über die näheren Umstände auf.- In Wahrheit haben sie ihm keinen Pinguin gekauft, sondern lediglich eine Patenschaft in seinem Namen übernommen. Und die Patenschaft dauert, so viel er weiß, ein Jahr. Die Torvalds-Familie versucht, das Konzept hinter Smores * zu verstehen. Zwischen zwei Graham Cracker werden ein Marshmallow und eine Hand voll Schokochips gelegt. Das Ganze wird eingepackt in Alufolie ein paar Minuten im Lagerfeuer geschmolzen.) Jemand schaut von seinem gerösteten Marshmallow hoch und macht den Fehler zu fragen, wie es überhaupt dazu kam, dass der Pinguin zum weltweiten Linux-Symbol wurde.*

„Der Pinguin war meine Idee“, sagt Tove. „Linus versuchte ein Symbol für Linux zu finden, weil die Leute ihn immer wieder fragten: ‚Wäre es nicht gut, ein Symbol zu haben?‘ Er dachte über die Logos nach, die er so kannte. Die Linux-Firmen hatten alle ihre eigenen Logos. Eines der Unternehmen verwendete ein pinkfarbened Dreieck als Symbol. Aber ich wusste, das war das internationale Symbol für Homosexuelle und sagte ihm, es sei schon vergeben. Dann sagte Linus, er hätte gern etwas Nettes, etwas Sympathisches.

Da dachte ich an Pinguine. Linus war in einem Zoo in Australien von einem Pinguin gebissen worden. Er streichelt gern Tiere und lässt seine Finger noch nicht einmal von Klapperschlangen und ähnlichem Zeug. Diese Pinguine im Zoo waren vielleicht dreißig Zentimeter groß, und er langte einfach in den Käfig, um sie zu streicheln. Er

** Anmerkung der Übersetzerin: Eine Leckerei für Kinder.*

bewegte seine Finger, als wären sie Fische. Die Pinguine kamen zu ihm her, bisßen ihn und stellten fest, dass er doch kein Fisch war. Aber er mochte sie trotzdem. Ich hatte den Eindruck, dass er nach diesem Erlebnis eine Leidenschaft für Pinguine entwickelte. Er ließ keine Gelegenheit aus, sich welche anzusehen.

Als er dann anfang, nach einem Logo zu suchen, sagte ich ‚Warum nimmst du keinen Pinguin, wo du doch Pinguine so gern magst?‘ Und er sagte: ‚Okay, ich werde darüber nachdenken.‘

An dieser Stelle schüttelt Linus, der vielleicht drei Leute von Tove entfernt sitzt, den Kopf.

„Nein, es war nicht ihre Idee“, sagt er. „Da liegt sie falsch.“

Das kommt überraschend. Linus und Tove sind nur selten unterschiedlicher Meinung. Tove hat die Mädchen, den Haushalt und einen berühmten Ehemann erstaunlich gut im Griff und hält mit ihren Karate Fähigkeiten sogar aufdringliche Journalisten auf Distanz. Linus trägt bereitwillig seinen Teil dazu bei, indem er gelegentlich die Wäsche faltet oder morgens den Cappuccino macht. Selbst den Stress einer zehnstündigen Autofahrt mit den immer neuen Bedürfnissen von zwei Kleinkindern bewältigen Linus und Tove spielend: Stellen Sie sich das eheliche Äquivalent eines durchdacht konstruierten skandinavischen Schlafsofas vor.

Wir hatten einen Schwachpunkt entdeckt.

Linus' Geschichte zufolge mag Tove zwar in einem frühen Stadium Pinguine vage erwähnt haben, aber erst nach einem Gespräch mit zwei hochrangigen Linux-Typen wurden die Geschöpfe aus dem Eis als offizielles Maskottchen des Betriebssystems ernsthaft in Erwägung gezogen.

Tove steuert ihre Deutung zu dieser Version bei. „Er kam zu dem Schluss, die Idee sei wohl doch nicht so gut, weil sie von mir stammte. Also zerbrach er sich weiter den Kopf über ein mögliches Logo. Dann waren wir mit Maddog und Henry Hall in Boston. Sie fingen an, über ein Logo zu reden. Ich sagte zu ihnen: ‚Wie wäre es mit einem Pinguin? Findet ihr nicht, dass ein Pinguin nett wäre?‘ Sie sagten ja. Ich denke, danach dachte Linus, die Idee könnte doch ganz gut sein.“

„Henry Hall sagte, er kenne einen Grafiker, der das Logo für ihn zeichnen könnte, aber dazu kam es nie. Ehe ich mich versah, hatte Linus die Internet-Gemeinde aufgefordert, ihm Bilder von Pinguinen

zu schicken.“ Er wählte die Version von Larry Ewing aus, einem Grafiker, der am Institute for Scientific Computing an der ABM University in Texas arbeitet.

Aber Linus wollte keinen x-beliebigen Pinguin. Sein Pinguin sollte glücklich aussehen, so als hätte er gerade eine Maß Bier genossen und den besten Sex seines Lebens gehabt. Und was noch wichtiger war: Er sollte unverwechselbar sein. Während alle anderen Pinguine einen schwarzen Schnabel und schwarze Füße haben, hat das Linux-Maskottchen deshalb einen orangefarbenen Schnabel und orangefarbene Füße, so dass es wie ein Pinguin aussieht, dessen Mutter eine Ente war. Als hätte Daisy Duck sich auf einer Antarktis-Kreuzfahrt vergessen und einen wilden One-Night-Stand mit einem einheimischen Federvieh gehabt.

Die Nachricht, dass ich mich entschieden hatte, für die Transmeta Corporation zu arbeiten, löste in **der Linux-Gemeinde die gleiche Reaktion aus wie die Nachricht, dass Tove und ich herausgefunden hatten, wie man ein Baby macht, und Ende 1996 Eltern werden würden.**

Als im Frühjahr durchsickerte, dass Tove schwanger war, wollten die sprachgewandteren Teilnehmer der Linux-User-Newsgroup wissen, **wie ich gedachte, die Anforderungen der Linux-Wartung mit denen einer Familie in Einklang zu bringen.** Ein paar Monate später, als bekannt wurde, dass ich (endlich) die Universität Helsinki verlassen würde, um für die geheimnisumwitterte Firma Transmeta im Silicon Valley zu arbeiten, entbrannte eine große, wortreiche Debatte, ob ich meiner Open-Source-Philosophie in einer notorisch kommerziellen Umgebung ebenso treu bleiben konnte wie an einer neutralen akademischen Institution. Die Firma würde teilweise von Microsoft-Mitbegründer Paul Allen **finanziert, protestierten die Leute, und manche verstiegen sich** sogar zu der Behauptung, das Ganze müsse ein perfider Plan sein, die Kontrolle über Linux zu übernehmen.

Ich kann verstehen, dass Leute, die der Linux-Szene loyal zugetan sind, sich Sorgen machen; ich wünschte nur, sie ließen die Kirche im Dorf. Tatsache ist: **Weder Patricias Geburt im Dezember 1996 (noch Danielas sechzehn Monate später noch Celestes achtundvierzig Monate später) noch mein Job bei Transmeta,** wo ich im Februar 1997 anfang, hat Linux geschadet. Für mich war immer klar gewesen: Wenn etwas meine Arbeit an Linux negativ beeinflussen würde, würde ich den notwendigen Schritt tun und alles an eine Person meines Vertrauens übertragen.

Aber ich greife mir vor.

Im Frühjahr 1996, als das Wetter endlich besser **wurde, erbrachte ich die letzten der für den Master-Abschluss erforderlichen Leistungsnachweise.** Ungefähr zu dieser Zeit bekam ich eine Nachricht von Peter Anvin, **jenem Linux-Fan, der drei Jahre zuvor die Online-Sammlung organisiert hatte, die mir bei der Abzahlung** meines ersten PCs geholfen hatte. Wie jeder andere, der die Linux-Newsgroups durchstreifte, wusste er, dass ich mein **Studium bald abschließen würde.** Er arbeitete seit etwa einem Jahr bei Transmeta und hatte bei seinem Chef **vorgefühlt, er kenne da diesen Typen in Finnland, der ein Gewinn für die Firma sein könnte.** Als er seine **Mutter in Schweden besuchte, nutzten wir die Gelegenheit für** ein kurzes Treffen. Er pries Transmeta in den höchsten Tönen, was ziemlich schwierig war, da die Firma sich im Tarn-Modus befand, und er mir nicht viel sagen konnte. Unter Programmierern ging lediglich das Gerücht um, Transmeta befasse sich mit der Entwicklung „programmierbarer Chips“. Auf jeden Fall war es super, Peter persönlich kennen zu lernen.

Eine Woche, nachdem er nach **Kalifornien zurückgekehrt war, schickte er mir eine Mail und fragte, ob ich kommen konnte. Das war völlig anders als die Erfahrung, die ich im Jahr zuvor mit Intel gemacht hatte, als ein Ingenieur mich für ein Praktikum einstellen wollte, das aber wegen des Papierkrams nie zustande kam.**

Es **würde mir Spaß machen, einfach mal wieder** nach Kalifornien zu fahren, dachte ich.

Das war das erste Vorstellungsgespräch meines Lebens. **Ich hatte keinen Lebenslauf.** Ich wusste nicht, was Transmeta eigentlich machte. Ich war in einem fremden Land.

Ich machte mir mehr Gedanken darüber, was es heißen würde, in die USA zu ziehen, als darüber, ob ich den **Job bekommen würde, so dass ich die Treffen mit den Transmeta-Leuten noch nicht einmal** als Vorstellungsgespräche empfand. Wichtiger schien es mir zu sein, herauszubekommen, was die Typen vorhatten. Es war eine **ziemlich seltsame Situation.**

Nach dem ersten Tag ging ich ins Hotel zurück, das Santa Clara Marriott gegenüber der Transmeta-Zentrale. **Vom Jetlag halb betäubt, fand ich das Gehörte sehr interessant, aber gleichzeitig hielt ich die Transmeta-Leute für übergeschnappt.** Damals hatte das Unternehmen noch **kein Silizium vorzuweisen.** Keine Hardware. Alles wurde über einen

Simulator erledigt, und die Vorführung, wie sich die Simulation nach Windows 3.11 bootete und Solitaire ausführte, vermochte mich nicht davon zu überzeugen, was das bringen sollte. Nach diesem ersten Tag fragte ich mich, ob wir nicht alle unsere Zeit verschwendeten. Ich weiß noch sehr genau, wie ich dachte: Vielleicht kommt bei der ganzen Sache nichts heraus - weder eine technologische Innovation für Transmeta noch ein Job für mich.

Ich schlief eine Nacht darüber, im wahrsten Sinne des Wortes. Nur dass ich eigentlich nicht viel Schlaf bekam. Ich lag im Bett und dachte über die Transmeta-Pläne nach. Dann malte ich mir aus, wie es wäre, im Garten eine Palme zu haben. Dann ließ ich mir durch den Kopf gehen, was ich auf dem Simulator gesehen hatte. Es war eine denkwürdige, unruhige Nacht, aber kein Vergleich zu den Angstträumen in Ede, die mir das Blut in den Adern gefrieren ließen.

Am Morgen war ich etwas aufgeregt. Am Ende des zweiten Tages war ich sehr aufgeregt. Und da ging der Stress los.

Bevor ich das Transmeta-Angebot akzeptierte, hatte ich mit verschiedenen Leuten gesprochen. Als sich herumsprach, dass ich den Job in Erwägung zog, erhielt ich eine Reihe anderer Angebote. In Finnland bekam ich ein Angebot von Tele, das für irgendeine Aufgabe Linux einsetzte. Durch Maddog erhielt ich ein Angebot von Digital. (Nichts für ungut, aber Boston im Winter ist nicht so viel besser als Helsinki im Winter. Okay, vielleicht doch.) Ich sprach mit ein paar Leuten von Red Hat. Sie boten mir einen Job an und sagten, sie würden mehr zahlen als Transmeta, obwohl sie keine Ahnung von der Höhe des Gehaltsangebots hatten, weil wir über Geld noch gar nicht gesprochen hatten. Die Red-Hat-Leute sagten, sie würden sogar die Aktienoptionen von Transmeta toppen, ganz gleich, wie hoch sie waren. Aber ich hatte kein Interesse daran, für eine Linux-Firma zu arbeiten - nicht einmal eine, die den Vorzug hatte, mitten in North Carolina zu liegen.

Am Ende erhielt ich fünf Jobangebote, ohne mich je offiziell nach einem Job umgesehen zu haben. Das Transmeta-Angebot war das bei weitem interessanteste.

Ich sagte zu. Das war ein komisches Gefühl. Als Nächstes teilte ich der Universität mit, dass ich kündigen würde. *Da* ging der Stress wirklich los. Ich war dabei, einen gigantischen Schritt zu wagen, von dem es kein Zurück gab. Wir bekamen ein neues Kind, übersiedelten in ein neues Land, ich verließ das sichere Nest der Universität Helsinki -

aber vorher musste ich noch meine Abschlussarbeit schreiben. Im Nachhinein betrachtet war es vermutlich eine gute Idee, dass wir all diese Veränderungen auf einmal hinter uns brachten. Trotzdem war es der helle Wahnsinn.

Es gab keine formelle Ankündigung (wozu auch?). Nur Buschtrommeln im Internet und die bereits erwähnte Debatte, ob ich in der Lage sein würde, Linux und dem Gedanken der Free Software auch in der bösen Welt von *Corporate America* und zwischen dem Wechseln von Windeln treu zu bleiben. Damals wurde Linux als Projekt gesehen, an dem Studenten arbeiten, nicht Leute, die es geschafft hatten. So gesehen war die Nervosität wohl verständlich.

Ich schrieb meine Abschlussarbeit an einem langen Wochenende und gab sie ab - Minuten, bevor ich Tove ins Krankenhaus brachte, wo 40 Stunden später Patricia zur Welt kam. Das war am 5. Dezember 1996. Vater zu sein erschien mir als die natürlichste Sache der Welt.

In den nächsten Wochen waren wir mit Patricia beschäftigt und fragten uns ständig, wann endlich die Unterlagen für unsere U.S.-Visa bearbeitet sein würden, was ewig dauerte. Wir überlegten uns, dass es hilfreich sein könnte, wenn wir verheiratet wären, deshalb gingen wir irgendwann im Januar - ich muss immer Tove nach dem Datum fragen - zum Standesamt, um offiziell die Ehe einzugehen. Wir hatten drei Gäste: Toves Eltern und meine Mutter. (Mein Vater war in Moskau.) Es war eine merkwürdige Zeit. Irgendwann verfrachteten wir fast alle unsere Habseligkeiten in die USA, ohne überhaupt zu wissen, wann wir fliegen konnten. Um uns von unseren Freunden zu verabschieden, gaben wir eine House-Cooling-Party, das Gegenstück einer House-Warrring-Party. In unserer kleinen, vor kurzem leer geräumten Ein-Zimmer-Wohnung drängten sich zwanzig Leute. Wir betranken uns, wie die finnische Tradition es verlangt.

Endlich kamen unsere Visa. Also bestiegen wir am 17. Februar 1997 die Morgenmaschine nach San Francisco. Ich erinnere mich noch an die Temperatur in Helsinki: minus 18 Grad Celsius. Ich erinnere mich an Toves Familie am Flughafen und die Tränen beim Abschied - sie stehen einander sehr nahe. Ich erinnere mich nicht daran, ob meine Familie da war oder nicht. Wahrscheinlich schon. Aber vielleicht auch nicht.

Dann landeten wir in den USA und bahnten uns mit dem Baby und zwei Katzen den Weg durch den Zoll. Peter Anvin war gekommen, um

uns in Empfang zu nehmen, und wir mieteten ein Auto für die Fahrt hinunter nach Santa Clara zu dem Apartment, das wir ein paar Monate zuvor gemietet hatten, als wir kurz zur Wohnungssuche nach Kalifornien geflogen waren. Alles wirkte surreal, vor allem der Temperaturunterschied von 21 Grad Celsius zu Finnland.

Unsere Sachen sollten erst in zwei Monaten ankommen. In der ersten Nacht schliefen wir auf unserer mitgebrachten Luftmatratze. Am nächsten Tag gingen wir los und kauften uns ein richtiges Bett. Bis unsere Möbel es nach USA schafften, musste Patricia in ihrem Kinderwagen schlafen. Tove war davon alles andere als begeistert, aber David meint, damit hätte sich sozusagen der Kreis geschlossen - in Anspielung auf meine ersten drei Lebensmonate, die ich in einem Wäschekorb verbrachte. Wir kochten nicht viel (das tun wir immer noch nicht) und wussten nicht, wohin wir zum Essen gehen sollten. Die meisten Mahlzeiten nahmen wir entweder in der Lebensmittelabteilung eines nahe gelegenen Einkaufszentrums oder in einem FastFood-Restaurant zu uns. Ich weiß noch, wie ich zu Tove sagte, dass wir ein paar neue Restaurants aufsuchen mussten.

Bedingt durch den Umzug und meine Einarbeitung bei Transmeta hatte ich in jenen ersten Monaten nicht viel Zeit, mich um Linux zu kümmern. Der neue Job nahm einen Großteil meiner Zeit in Anspruch, und die Stunden nach der Arbeit verbrachte ich mit Tove und Patricia, und wir versuchten, die neue Umgebung kennen zu lernen. Mein Leben war damals ziemlich hektisch. Wir hatten absolut kein Geld. Ich hatte zwar ein phantastisches Gehalt, aber das ging komplett für den Kauf von Möbeln drauf. Unsere Autos zu kaufen, war ein harter Kampf, weil wir keine Möglichkeit hatten, unsere Kreditwürdigkeit nachzuweisen. Wir mussten sogar darum kämpfen, nachzuweisen, dass wir in der Lage sein würden, die Telefonrechnung zu bezahlen.

Mein Computer befand sich auf einem Schiff, das sich langsam um das Horn von Afrika herumbewegte. Es war das erste Mal, dass ich über längere Zeit hinweg nichts von mir im Netz hören ließ, und meine Abwesenheit sorgte für Beunruhigung. Der Tenor war, okay, jetzt ist er in die Industrie gegangen ...

Ohne Umschweife wurde ich gefragt: Heißt das, dass Linux als freies System stirbt? Ich erklärte, dass mein Vertrag mit Transmeta mir erlaubte, weiterhin für Linux da zu sein. Und dass ich nicht von der

Bildfläche verschwinden würde. (Ich wusste nicht, wie ich vermitteln sollte, dass ich einfach mal eine Verschnaufpause brauchte.)

Leben in Transmetaland.

Die besondere Schwierigkeit, den Leuten zu erklären, dass der Umzug in die USA und der Eintritt in die kommerzielle Welt mich nicht verändern würden, ergab sich daraus, dass Transmeta die verschwiegenste Firma war, die man sich vorstellen konnte. Es gab eine einzige Regel darüber, was man sagen durfte, und sie lautete ganz einfach: „Sag nichts.“ Unter Linux-Anhängern warf das die Frage auf, welcher seltsamen Sekte ich mich da angeschlossen hatte und ob ich je zurückkommen würde. Ich konnte nicht einmal meiner Mutter erzählen, was ich dort eigentlich machte - nicht dass es sie interessiert hätte.

Dabei war das, was ich bei Transmeta tat, keineswegs ungewöhnlich. Als Erstes löste ich tatsächlich ein paar der Linux-Probleme, die bei Transmeta aufgetreten waren. Das neue Unternehmen setzte eine Menge Multiprozessor-Maschinen unter Linux ein. Ich hatte persönlich nie an den Linux-SMP-Fragen (Symmetrisches Multiprocessing) mitgearbeitet, und es stellte sich heraus, dass vieles nicht so lief, wie es hätte laufen sollen. Ich empfand das als persönlichen Affront und musste das Problem natürlich lösen.

Aber meine eigentliche Arbeit bestand darin, Teil des Transmeta-Softball-Teams zu sein.

Oh, ich meine natürlich des Software-Teams. Wir hatten nicht oft Gelegenheit, Softball zu spielen. Keine der Mannschaften im Silicon Valley wäre gegen uns angetreten, es sei denn, wir hätten uns bereit erklärt, ihnen zu sagen, was wir eigentlich machten.

Ich weiß nicht, inwiefern Transmeta den Leuten ein Begriff ist. Tatsächlich befinden wir uns, während ich das schreibe, in unserer Schweigephase vor (bitte, kaufen Sie um Himmels willen unsere Aktien) dem Börsengang, und die Heimlichtuerei hat ein Ende, auch wenn wir uns schon wieder im Tarn-Modus befinden, diesmal allerdings wegen der Auflagen der US-Bundesbehörde Securities and Exchange. Es steht zu hoffen, dass bei Erscheinen dieses Buches jeder Mensch und sein Hund von Transmeta gehört und (*unterschwellige Botschaft: AKTIEN*) einen oder mehrere unserer Prozessoren gekauft hat. Das nämlich machen wir bei Transmeta: Prozessoren. Hardware.

Aber Transmeta macht mehr als nur Hardware. Glücklicherweise, denn, ehrlich gesagt, ich könnte einen Transistor nicht von einer

Diode unterscheiden, selbst wenn man mich schlagen würde. Was Transmeta macht, ist *einfache* Hardware, die durch intelligente Software einen einfachen Prozessor als sehr viel mehr erscheinen lässt, als er eigentlich ist - nämlich als einen ganz normalen Intel-kompatiblen x86-Prozessor. Und weil die Hardware kleiner und einfacher ist, hat sie weniger Transistoren, was bedeutet, dass sie weniger Energie verbraucht, was, wie jeder weiß, in einer mobilen Welt immer wichtiger wird. Wegen dieser intelligenten Software gibt es bei Transmeta ein ziemlich großes Softwareteam - und mich.

Das kam mir alles sehr gelegen. Eine Nicht-Linux-Firma, die etwas technisch ziemlich Interessantes machte (Understatement des Jahres -ich kenne immer noch keine andere Firma, die je ernsthaft versucht hat, das Gleiche wie Transmeta zu machen.) Noch dazu in einem Bereich, mit dem ich mich bestens auskannte: maschinennahe Programmierung der ziemlich esoterischen 80x86er-Prozessorfamilie. Wie Sie sich zweifellos erinnern, wurde ja das ganze Linux-Projekt erst durch meinen Ehrgeiz auf den Weg gebracht, den Prozessor meines ersten eigenen PCs zu erforschen.

Die Tatsache, dass Transmeta keine Linux-Firma war, war mir ebenfalls wichtig. Verstehen Sie mich nicht falsch: Es machte mir Spaß, die Linux-Probleme bei Transmeta zu lösen, und ich war an mehreren internen Linux-Projekten beteiligt. (Tatsächlich ist es heute wahrscheinlich unmöglich, ein ernsthaftes Technologieunternehmen ohne solche Projekte zu finden.) Aber für Transmeta war Linux sekundär. Und genau das wollte ich. Ich konnte weiter an Linux arbeiten, aber ich hatte nicht das Gefühl, technische Kompromisse eingehen zu müssen, die mehr den Unternehmenszielen als Linux selbst zugute kamen. Ich konnte Linux weiter als Hobby betrachten, bei dem nur die Technik zählte und nichts anderes meine Entscheidungen beeinflusste.

Also arbeitete ich tagsüber für Transmeta. Ich schrieb und wartete den „x86er Interpreter“, den wir immer noch nutzen (auch wenn heute andere für seine Wartung zuständig sind). Der Interpreter ist im Prinzip der Teil der Transmeta-Software, der sich die Intel-Befehle nacheinander ansieht und sie ausführt (d.h. er „interpretiert“ befehlsweise die Sprache der 80x86er-Architektur.) Später übernahm ich dann andere Aufgaben, aber erst einmal betrat ich die fremde und wunderbare Welt der Hardware-Emulation.

Nachts schlief ich.

Meine Abmachung mit Transmeta war klar: Ich hatte die vage Zusage bekommen, auch während der Arbeitszeit an Linux arbeiten zu können. Glauben Sie mir, dieses Vorrecht nutzte ich voll aus. Viele Leute setzen auf Überstunden und Doppel-, Dreifach- oder sogar Vierfachschieben. Ich nicht. Weder Transmeta noch Linux haben mich je um meine Nachtruhe gebracht. Tatsächlich glaube ich, wenn Sie die ehrliche Wahrheit wissen wollen, fest an die segensreiche Kraft des Schlafes. Manche Leute halten diese Einstellung für ein Zeichen von Faulheit, aber auf sie würde ich am liebsten mein Kissen werfen. Ich habe für meine Schlafgewohnheiten einen guten Grund, und dazu stehe ich: Du verlierst vielleicht ein paar produktive Tagesstunden, wenn du, ach, sagen wir ruhig, zehn Stunden am Tag schläfst, aber in den paar Stunden, in denen du wach bist, bist du voll da, und dein Gehirn läuft wie ein Sechszylinder. Oder ein Vierzylinder. Was auch immer

Willkommen im Silicon Valley. Zu meinen ersten Verpflichtungen nach der Landung in dieser fremden Galaxie gehörte es, die Stars zu treffen.

Ich erhielt eine E-Mail von Steve Jobs Sekretärin, dass er mich gern kennen lernen würde und ob ich eine Stunde oder zwei erübrigen könnte. Da ich nicht wusste, worum es ging, sagte ich, ja, klar.

Das Treffen fand in der Apple-Zentrale am Infinity Loop Drive statt. Jobs hatte seinen Chefentwickler dabei, Avie Tevanian. Apple hatte damals gerade mit der Arbeit an OS X begonnen, dem Unix-basierten Betriebssystem, das erst im September 2000 freigegeben wurde. Wir hielten uns nicht lange mit Formalitäten auf. Jobs klärte mich ohne große Vorreden darüber auf, dass es im Desktop-Bereich nur zwei Player gab, Microsoft und Apple, und dass ich seiner Meinung nach für Linux nichts Besseres tun könnte, als mit Apple ins Bett zu gehen und zu versuchen, die Open-Source-Leute hinter Mac OS X zu bringen.

Ich hielt nur an mich, weil ich mehr über das neue Betriebssystem erfahren wollte. Mac OS X setzt auf Mach auf, dem Microkernel, der an der Carnegie Mellon University entwickelt worden war. Mitte der 1990er-Jahre rechnete man damit, dass Mach sich als das ultimative Betriebssystem erweisen würde, und eine Menge Leute interessierten sich dafür. Tatsächlich nutzten IBM und Apple Mach als Basis für ihr unglückseliges gemeinsames Betriebssystemprojekt Taligent.

Jobs machte ein großes Trara um die Tatsache, dass der Quellcode des maschinennahen Kernels von Mach frei zugänglich ist. Über den Makel dieser Konstellation schwieg er sich dagegen aus: Was nutzt es schon, dass der Betriebssystemkern, der wirklich maschinennahe Teil,

frei zugänglich ist, wenn die darüber liegende Mac-Schicht nicht frei zugänglich ist?

Er konnte nicht wissen, dass ich persönlich Mach nicht besonders schätze. Ehrlich gesagt, halte ich es für Schwachsinn. Es weist alle nur denkbaren Entwurfsfehler auf und steuert selbst noch ein paar neue bei. Eines der Argumente gegen Microkernels war immer die Leistung gewesen. Deshalb versuchten eine Reihe von Leuten, in Forschungsprojekten festzustellen, wie sich die Leistung von Microkernels erhöhen lässt. Alle daraus resultierenden Empfehlungen flossen in den Mach-Kernel ein. Das führte dazu, dass er zu einem sehr komplizierten System mit eigenen Regeln wurde. Und seine Leistung lässt *weiterhin* zu wünschen übrig.

Avie Tevanian gehörte bereits zu den Mach-Leuten, als das Betriebssystem noch ein universitäres Projekt war. Es war interessant, die Themen zu diskutieren, die ihm und Steve am Herzen lagen. Gleichzeitig vertraten wir in technischen Fragen grundsätzlich andere Meinungen. Ich sah wirklich nicht, warum die Open-Source-Bewegung oder Linux-Leute sich daran beteiligen sollten. Klar, ich konnte verstehen, warum sie mehr Open-Source-Entwickler für ihr System interessieren wollten; sie sahen die Eigendynamik, die Linux entwickelt hatte. Aber ich glaube nicht, dass sie sich wirklich eine Vorstellung davon machten. Ich glaube nicht, dass Jobs erkannte, dass Linux eines Tages mehr Anwender haben würde als Apple, auch wenn sein Anwenderkreis ein ganz anderer ist. Ich glaube nicht, dass Steve Linux heute so vorschnell wie vor drei Jahren als Desktop-System abtun würde.

Ich erklärte, was mir an Mach missfiel. Aus verständlichen Gründen kam das bei meinen Gesprächspartnern nicht besonders gut an. Mit Sicherheit hatten sie die Argumente schon vorher gehört. Offensichtlich war ich stark auf Linux fixiert, und Tevanian war stark auf Mach fixiert. Es war interessant, ihre Diskussion über ein paar technische Probleme mitzubekommen. Zu den drängendsten Problemen, die ich sehen konnte, gehörte die Unterstützung der alten Mac-Anwendungen in dem neuen Betriebssystem. Sie hatten vor, das ganze alte Zeug über eine Kompatibilitätsschicht anzubinden. Alle alten Mac-Anwendungen sollten in einem aufgepfropften Prozess laufen. Aber eine der Hauptunzulänglichkeiten des alten Mac ist ein fehlender Speicher-schutzmechanismus, und die geplante Lösung würde nichts zur Über-

windung dieses Problems beitragen. Nur die neuen Mac-Anwendungen würden über einen Speicherschutzmechanismus verfügen. Für mich machte das Ganze keinen Sinn.

Wir betrachteten die Welt mit völlig unterschiedlichen Augen. Steve war Steve. Genau so wie die Presse ihn darstellt. Er interessierte sich für seine eigenen Ziele und vor allem für das Marketing. Ich interessierte mich für die technische Seite, und seine Ziele oder Argumente waren mir ziemlich egal. Sein Hauptargument war, dass ich mich mit Apple zusammentun sollte, um in den Desktop-Markt einzudringen. Meine Reaktion darauf: Was sollte mir daran liegen? Welches Interesse sollte ich an der Apple-Geschichte haben? Ich fand an Apple nichts interessant. Und es gehörte auch nicht zu meinen Lebenszielen, den Desktop-Markt zu übernehmen. (Klar, das wird geschehen, aber es ist nie mein Ziel gewesen.)

Er brachte nicht sehr viele Argumente vor. Er setzte mein Interesse einfach voraus. Er hatte keine Ahnung. Er konnte sich nicht vorstellen, dass weite Teile der menschlichen Rasse nicht das geringste Verlangen danach hatten, den Marktanteil des Mac zu vergrößern. Ich glaube, er war ehrlich überrascht, wie wenig es mich interessierte, wie groß der Marktanteil des Mac war - und wie wenig es mich interessiert, wie groß der Marktanteil von Microsoft ist. Und ich kann ihm keinen Vorwurf daraus machen, dass er nicht im Voraus wusste, wie wenig ich dem Mach-Kernel abgewinnen konnte.

Aber obwohl ich fast keine seiner Meinungen teilte, war er mir irgendwie sympathisch.

Dann traf ich das erste Mal Bill Joy. Ich stand auf und ging.

Okay, fairerweise muss ich dazu sagen, dass ich bei unserer ersten Begegnung nicht wusste, wer er war. Es war bei der ersten Jini-Vorstellung. Jini, eine Java-Erweiterung, ist eine von Sun-Microsystems entwickelte Sprache für interagierende Agenten. Sie unterstützt die nahtlose Vernetzung völlig unterschiedlicher Systeme. Wenn man zum Beispiel einen Jini-fähigen Drucker hat, können alle anderen Geräte im gleichen Netz, die Jini sprechen, den Drucker automatisch nutzen.

Sun hatte im Vorfeld der offiziellen Ankündigung mich und ein Dutzend anderer Open-Source-Typen sowie ein paar technische Leute zu einem privaten Briefing eingeladen, das während der Java World in

einem Hotelzimmer im Stadtzentrum von San Jose stattfand. Der Grund, aus dem wir eingeladen wurden: Sie hatten Jini als Open-Source-Software entwickelt oder als das, was Sun Microsystems dafür hält.

Ich ging hin und mir war klar, dass Bill Joy auch dort sein würde. Er war die Schlüsselperson hinter BSD Unix gewesen und später als Chefwissenschaftler zu Sun gegangen. Ich war ihm bisher noch nicht begegnet. Er kam einfach zu mir herüber und sagte, er sei Bill Joy, und irgendwie reagierte ich nicht darauf. Ich war nicht gekommen, um ihn zu treffen, sondern um zu sehen, wie Sun dem Open-Source-Gedanken gegenüberstand und ihn verwirklichen wollte. Ein paar Minuten danach erklärte Bill selbst die Gründe, warum sie Jini frei zugänglich machen wollten, und wir bekamen eine kleine Demo des Systems zu sehen.

Dann fingen sie an, ihre Lizenzpolitik zu erklären. Sie war grauenhaft. Einfach bescheuert. Im Prinzip lief sie auf die Tatsache hinaus, dass schon bei einer halb-kommerziellen Nutzung von Open Source keine Rede mehr sein konnte. Ich hielt das ganze für eine völlig idiotische Idee. Am meisten regte mich auf, dass sie ihre Aufgeschlossenheit für Open Source auf der Einladung so herausgestrichen hatten. Open Source schien für sie zu bedeuten, dass sie dir erlaubten, den Quellcode zu lesen. Aber sobald du den Code verändern oder in deine Infrastruktur einbetten wolltest, musstest du von Sun eine Lizenz erwerben. Angenommen, jemand bei Red Hat wollte die jüngste Red-Hat-CD-Version von Linux Jini-fähig machen, dann würde die Firma von Sun eine Lizenz der Jini-Technologie erwerben müssen.

Ich stellte ein paar Fragen, um mich zu vergewissern, dass ich das richtig verstanden hatte.

Dann stand ich auf und ging.

Ich war so angewidert, dass sie die Leute mit der Behauptung hergelockt hatten, eine Open-Source-Software vorzustellen, dass ich, als ich die Sache durchschaute, sinngemäß sagte „Vergesst es. Kein Interesse" und sie sitzen ließ.

Ich hatte den Eindruck, sie wollten mich da haben, um mich einfach mal zu informieren, und wenn ich mich begeistert gezeigt hätte, hätten sie gern ein Presse-Statement oder so von mir gehabt. Dieser Plan ging nicht auf. Vielleicht lernten sie etwas daraus. Anscheinend ließen sie

sich später davon überzeugen, Star Office als Open-Source-Software anzubieten. Vermutlich ist alles nur eine Frage der Zeit. So viel ich weiß, wurde die Veranstaltung fortgesetzt, ein gemeinsames Abendessen schloss sich an, und alle anderen harrten bis zum Schluss aus.

Meine zweite Begegnung mit Bill Joy verlief erheblich erfreulicher. Etwa eineinhalb Jahre später lud er mich ein, mit ihm Sushi zu essen. Seine Sekretärin rief mich an, um einen Termin zu vereinbaren. Bill lebt und arbeitet in Colorado und verbringt anscheinend eine Woche jedes Monats im Silicon Valley. Wir gingen zu Fuki Sushi in Palo Alto. Es ist eines der besseren Sushi-Restaurants im Silicon Valley. Natürlich reicht es nicht an Blowfish Sushi in San Francisco heran, wo man nonstop japanische Animationen zu sehen bekommt, oder an Tokyo Go Go im Mission-Viertel mit seiner angesagten Szene oder an Sushi Ran in Sausalito mit seinen wichtigen Stammgästen oder an Seto Sushi in Sunnyvale, wo es das beste pikante Thunfisch-Sushi überhaupt gibt.

Okay, wir saßen also bei Fuki Sushi, und es war ganz lustig, weil Bill versuchte, echtes Wasabi zu bekommen. Ich wusste damals noch nicht, dass in den meisten japanischen Restaurants in den USA das, was als Wasabi serviert wird, nur gefärbter Meerrettich ist. Es stellte sich heraus, dass die Wasabi-Pflanze ausschließlich in japanischen Flüssen wächst und sich deshalb nur schwer kommerziell anbauen lässt. Bill versuchte, das der Kellnerin zu erklären, aber sie verstand nicht, worauf er hinaus wollte. Obwohl sie Japanerin war, war Wasabi für sie gleich Wasabi. Er bat sie, in der Küche nachzufragen.

Das verbale Schlagabtausch mit ihm machte Spaß. Unser Treffen war rein gesellschaftlicher Natur. Er vermittelte mir zwar, dass ich es ihm nur zu sagen brauchte, wenn ich für Sun arbeiten wollte, und er würde etwas arrangieren. Aber das war nicht die Hauptsache. Das Gespräch mit ihm war eher eine Gelegenheit, über Fachliches zu reden. Er erinnerte sich daran zurück, wie er fünf Jahre lang BSD Unix gepflegt hatte, und wusste es inzwischen überaus zu schätzen, dass Sun sich um die kommerzielle Seite kümmerte. Er sprach darüber, wie wichtig ihm die kommerzielle Unterstützung war, die ein Unternehmen wie Sun bieten konnte. Es war witzig und interessant, ihn über die Anfangstage von Unix reden zu hören. Und es störte mich nicht im Geringsten, dass wir nie echtes Wasabi zu kosten bekamen. Ich erinnere mich genau daran, wie ich dachte, dass er von den Berühmtheiten des Silicon Val-

ley, die ich bisher kennen gelernt hatte, wahrscheinlich der netteste und interessanteste war.

Zeitsprung: Drei Jahre später stoße ich in der Zeitschrift *Wired* auf einen unheimlich negativen Artikel über Technologie von Bill mit dem Titel „Die Zukunft braucht uns nicht“. Irgendwie war ich enttäuscht. Es ist offensichtlich, dass die Zukunft uns nicht braucht. Aber das hätte er nicht so negativ ausdrücken müssen.

Ich möchte Bills Artikel nicht Zeile für Zeile auseinander nehmen, aber ich bin überzeugt davon, dass der Menschheit nichts Traurigeres passieren könnte, als ewig im gleichen Trott weiterzumachen, statt sich zu dehnen und zu entfalten. Bill schien zu denken, Fortschritte wie Genmanipulation gefährdeten unser Menschsein. Alle denken dauernd, etwas, das anders ist als wir, sei unmenschlich. Das liegt daran, dass das, was wir jetzt sind, unsere Vorstellung des Menschseins prägt. Aber wenn wir laufend mit neuen Entwicklungen, wie immer sie auch aussehen mögen, mitwachsen, werden wir in 10 000 Jahren nicht Menschen nach den Standards von heute sein. Wir werden einfach eine andere Art von Menschen sein.

Seinem Artikel nach zu schließen hat Bill davor Angst. Meiner Meinung nach ist es unnatürlich - und sinnlos -, die Evolution aufhalten zu wollen. Statt nach zwei geeigneten Hunden zu suchen, die den gewünschten Nachwuchs produzieren, werden wir uns zweifellos der Genetik bedienen. Es scheint unvermeidlich, dass es mit Menschen eines Tages genauso sein wird. Meiner Meinung nach ist es besser, die menschliche Rasse durch Genetik zu verändern, als den Status quo festzuzementieren. Ich denke einfach, von einer höheren Warte aus betrachtet wäre es unendlich viel interessanter, eine fortgesetzte Evolution nicht nur des Menschen, sondern auch der Gesellschaft sicherzustellen, ganz gleich, welche Richtung sie einschlägt. Du kannst der Technik nicht Einhalt gebieten und du kannst unserem fortschreitenden Wissen über den Lauf des Universums und den Entwurf des Menschen nicht Einhalt gebieten. Das alles geht so schnell, dass manche Leute wie Bill Joy davor zurückschrecken. Aber ich sehe es als Teil unserer natürlichen Evolution. Ich teile Bill Joys Ansichten, wie wir der Zukunft begegnen sollen, ebenso wenig wie sein Open-Source-Konzept. Ich teilte auch nicht Steve Jobs Auffassungen über Technologie. Das klingt, als hätte ich meine ersten Jahre im Silicon Valley in besserwisserischer Sturheit ver-

bracht, aber so war es nicht. Ich habe viel programmiert, ich bin mit Patricia in den Streichelzoo gegangen und ich habe ganz allgemein meinen Horizont erweitert - und zum Beispiel die schreckliche Wahrheit über Wasabi erfahren.

V.

Über Nacht zum Erfolg.

Schauen Sie manchmal bei Advocacy-Newsgroups vorbei? Sie scheinen ihre Existenzberechtigung ausschließlich daraus zu beziehen, sich **für** eine Sache einzusetzen, indem sie eine andere heruntermachen. Wenn Sie dort hingehen, werden Sie nichts als unsinnige Behauptungen wie „Mein System ist besser als dein System“ finden. Es ist eine spezielle Form der Online-Masturbation.

Ich erwähne Advocacy-Newsgroups aus einem einzigen Grund: Trotz ihrer Absurdität geben sie einen Hinweis auf die neuesten Entwicklungen. **Als die Industrie beschloss, Linux sei der Darling der Betriebssysteme, wurde die wachsende kommerzielle Unterstützung zum Beispiel anfangs nicht in der Presse oder an der Kasse von Fry's Electronics diskutiert, sondern in Advocacy-Newsgroups.**

Zurück zur Chronologie. Im Frühjahr 1998 trat eine dritte Blondine in mein Leben: am 16. April kam Daniela Yolanda Torvalds zur Welt und war damit die erste Torvalds, die die amerikanische Staatsbürgerschaft besaß. Sie und Patricia sind 16 Monate auseinander, wie Sara und ich. Aber ich garantiere Ihnen, sie werden nicht so ineinander **verkeilt** sein, **wie meine Schwester und ich** es als Kinder waren. Toves mildernder Einfluss wird das zu verhindern wissen. Oder ihre Karatefähigkeiten.

Wenige Wochen nach Danielas Geburt erhielt die Open-SourceGemeinde - die damals noch Free-Software-Gemeinde hieß - ihren bis **dahin größten Auftrieb. Das Unternehmen Netscape kündigte an, es würde den Quellcode für seine Browser-Technologie** in einem Projekt namens Mozilla frei zugänglich machen. Einerseits fanden die Leute in den Newsgroups die Nachricht aufregend, weil sie die Sichtbarkeit der

Open-Source-Bewegung erhöhte. Aber gleichzeitig reagierten viele Leute, darunter auch ich, ziemlich nervös darauf. Das Unternehmen steckte damals in Schwierigkeiten, größtenteils wegen Microsoft, und die Preisgabe des Browsers wurde auch als Verzweiflungstat wahrgenommen. (Ironischerweise hatte der Browser seine Wurzeln im Open Source-Bereich. Er war ursprünglich ein Projekt an der Universität Illinois gewesen.)

Die Leute in den Newsgroups äußerten sich besorgt darüber, dass Neucap es vermessen und den Ruf der Open-Source-Bewegung schädigen würde. Von jetzt an würde es zwei namhafte Open-Source Projekte geben - Netscape und Linux -, und man befürchtete, ein Scheitern von Netscape, dem bekannteren der beiden, könnte auch das Ansehen von Linux beeinträchtigen.

Und Netscape scheiterte tatsächlich, jedenfalls im Großen und Ganzen. Das Unternehmen hatte Schwierigkeiten damit, Open-Source Entwickler auf Dauer für das Projekt zu interessieren. Der Codeumfang war riesig, und die einzigen Leute, die sich in diesen Code hineinwählen konnten, waren Neucap-Leute.

Neben seiner Größe war das Projekt auch dadurch gehandicapt, dass Neucap sich nicht in der Lage sah, alles als Open-Source-Code zur Verfügung zu stellen, und nur die Entwicklungsversion freigab, die zu diesem Zeitpunkt ziemlich kaputt war. Das Unternehmen konnte den Browser nicht unter die GPL stellen, weil ihm der Code nicht komplett gehörte - der Java-Teil zum Beispiel war von Sun lizenziert. Die Netscape-Lizenz stieß in den Newsgroups nicht nur auf Beifall. Insgesamt war sie ziemlich gemäßigt, aber wenn du jemand wie Richard Stallman bist, hast du mit Maß nicht viel am Hut.

Ich fand es wunderbar, dass Neucap diesen Schritt wagte, aber ich betrachtete ihn nicht als persönliche Leistung. Ich weiß noch, dass Eric Raymond die Sache wirklich persönlich nahm und sich unglaublich darüber freute. Sein Artikel „The Cathedral and the Bazaar“, der die

Philosophie und Geschichte der Open-Source-Bewegung ausgezeichnet erklärte, war im Jahr davor erschienen und wurde als einer der Gründe für die Neucap-Entscheidung genannt. Eric leistete dem Open-Source-Gedanken aktiv Vorschub. Er war mehrere Male bei Netscape gewesen und hatte versucht, das Unternehmen davon zu überzeugen, den Browser frei zugänglich zu machen. Ich bin nur einmal da gewesen. Tatsächlich hatte Eric eine Reihe von Unternehmen

besucht, um ihnen die Frohbotschaft des Open-Source-Gedankens zu verkünden. Ich interessierte mich für die Technik, aber ich wollte niemanden bekehren.

Binnen 24 Stunden nach der Freigabe von Mozilla entwickelte ein australisches Team, das sich als Mozilla Crypto Group bezeichnete, das Kryptographiemodul. Damals war es nicht-amerikanischen Staatsbürgern verboten, Verschlüsselungstechniken einzusetzen, die auf U.S.-amerikanischem Boden entstanden waren. Nun hatte plötzlich jemand aus Australien die Arbeit geleistet, so dass nichtamerikanische Staatsbürger sie nutzen konnten. Aber die Sache hatte einen Haken. Angesichts der geltenden Exportbeschränkungen Mozilla-Projekt den australischen Code nicht übernehmen. Wenn er erst einmal in den USA angelangt war, würde es nicht mehr möglich sein, ihn zu reexportieren. Das hieß, einer der ersten Erfolge des großen Neucap-Experiments konnte nicht Teil von Mozilla werden.

Wir waren alle besorgt, weil die Medien sich ausführlich mit Netscape beschäftigten. Deshalb gingen die Leute in diesem ersten Jahr wie auf Eiern. Niemand wollte etwas Negatives über Neucap sagen, aus Angst, den Open-Source-Gedanken als solchen bei der Presse in Misskredit zu bringen und andere Unternehmen zu

Aber zwei Monate nach der Neucap-Ankündigung kam Sun Microsystems ins Spiel und gab bekannt, das Unternehmen würde als erster großer Hardwarehersteller Linux International beitreten und Linux auf seinen Servern unterstützen. Ungeachtet seiner wenig beeindruckenden Lizenzpolitik für das Jini-Projekt hatte Sun nun beschlossen, dass es sich lohnte, Linux ernst zu nehmen. Die Newsgroups konnten sich selbst nicht genug auf die Schulter klopfen. Mit Sun an Bord fanden Linux-Entwicklungen ihren Weg vom Internet in die Fachpresse. Plötzlich waren Außenseiter an Linux interessiert, aber vor allem Außenseiter mit technischem
Dann kam IBM.

IBM galt als ziemlich schwerfällig, deshalb kam es für alle überraschend, als das Unternehmen im Juni ankündigte, es würde Apache, die beliebteste kommerzielle Linux-Version auf Web-Servern, vertreiben und unterstützen. Man konnte Apache unter AIX, der Unix-Variante von IBM, ausführen. Wahrscheinlich nutzten viele Leute, die IBM-Rechner kauften, diese Möglichkeit, und so wurde IBM auf die aufmerksam. Jemandem muss aufgefallen sein, dass die meisten

dieser Servermaschinen irgendwann Apache einsetzen. Deshalb rechnete IBM sich aus, mehr Server verkaufen zu können, wenn das Unternehmen intern das Know-how besaß, solche Kunden zu unterstützen. Vielleicht nahmen sie sich auch das Feedback von Kunden zu Herzen, die kundtaten, dass sie IBM-Rechner kaufen, aber Apache darauf ausführen würden.

Es ist relativ einfach, Linux auf einem Computer zu installieren. Aber für die meisten Unternehmen war von jeher die Frage entscheidend: Wem geben wir die Schuld, wenn etwas schief läuft? Obwohl natürlich Linux-Unternehmen wie Red Hat den Support übernehmen, bedeutete es einen psychologischen Schritt, den Kunden zu versichern, IBM würde für sie da sein. Als IBM anfing, sich dem Open-Source-Gedanken zu öffnen, argwöhnten viele Leute, das sei nur ein Lippenbekenntnis. Es stellte sich jedoch heraus, dass das nicht der Fall sein sollte. IBM hielt die Zehen ins Wasser, indem es Linux auf seinen Servern installierte und unterstützte, und wagte sich dann immer weiter vor. Als Nächstes folgten die kleinen PC-Server. Dann die regulären PCs. Und dann, dann kamen die Laptops. Das Unternehmen hat angekündigt, dieses Jahr eine Milliarde Dollar in Linux zu investieren.

IBM erledigte einen Großteil dieser Arbeit selbst. Ich denke, an Linux gefiel ihnen unter anderem, dass sie damit tun konnten, was sie wollten, ohne sich mit Lizenzfragen herumschlagen zu müssen. Das Unternehmen hatte eine Reihe unangenehmer Erfahrungen hinter sich. IBM war von Microsoft über den Tisch gezogen worden, nachdem die beiden Unternehmen gemeinsam das Betriebssystem OS/2 entwickelt hatten, ein, wie wir heute wissen, mit Steroiden voll gepumptes Windows. Microsoft unterließ es, OS/2 zu unterstützen, weil es kein Interesse daran hatte, sich den Markt teilen zu müssen. Auf der Microsoft-Seite kam immerhin NT dabei heraus. Für IBM aber haben sich die Milliarden von Dollar, die das Unternehmen in OS/2 pumpte, nie ausgezahlt. Außerdem musste IBM sich mit der Lizenzierung von Java herumschlagen. Ich denke, sie waren einfach froh, dass ihnen all diese Mühen bei Linux erspart blieben.

IBM war zweifellos der größte Coup. Und die Newsgroups reagierten mit ungeteiltem Enthusiasmus darauf - nicht mit der Paranoia, die die Neucapex-Ankündigung auslöste, oder dem aufgeheizten Anti-Kommerzialismus, der die Linux-Anhänger phasenweise (okay: häufig) entzweite.

Im Juli des gleichen Jahres gab Informix bekannt, es würde seine Datenbanken auf Linux portieren, so dass man, selbst wenn ein Computer unter Linux lief, eine Informix-Datenbank darauf nutzen konnte. Das war damals keine große Sache. Das Unternehmen hatte mit finanziellen Schwierigkeiten zu kämpfen, aber es gehörte nach wie vor zu den Top 3 unter den Datenbank Anbietern. Die Linux-Anhänger waren von dieser Entwicklung recht angetan und schrieben frohlockende Beiträge in den Linux-Advocacy-Groups.

Innerhalb weniger Wochen zog aus dem Nichts auftauchend Oracle nach. Oracle dominierte den Datenbankbereich. Schon lange vor der Ankündigung gab es Gerüchte (in den Newsgroups), das Unternehmen verfüge über interne Portierungen auf Linux. Und da Oracle und Unix-Server praktisch Synonyme sind, war die Sache für Linux wirklich kein Quantensprung. Aber wenn man die Newsgroups verfolgte, hätte man glauben können, für Linux wären große Zeiten angebrochen. Die Oracle-Ankündigung hatte psychologisch eine riesige Wirkung, obwohl sie technisch so gut wie nichts brachte.

Wie die IBM-Ankündigung wurde die Bekanntgabe Oracles nicht nur von der Linux-Szene, sondern auch von den Entscheidungsträgern im Management oder - kürzer - den „Anzugträgern“ als einschneidender Schritt empfunden. Künftig würden sie sich nicht mehr darauf herausreden können, Linux nicht einsetzen zu können, weil ihr Geschäft von Datenbanken abhing.

So erfreulich diese Entwicklungen auch sein mochten, sie veränderten mein Leben nicht. Wir hatten zwei bezaubernde Kinder zu versorgen. Die meisten meiner Nicht-Familien-Stunden verbrachte ich mit der Pflege von Linux, sowohl zu Hause als auch im Büro. Um keine einzelne Linux-Version zu begünstigen, setzte ich im Büro Red Hat und zu Hause SuSe, eine europäische Version, ein. Irgendwann hatte ich das Gefühl, dass ich zu wenig Bewegung bekam, und so beschloss ich, künftig die sechs Meilen zwischen unserer Wohnung und der Transmeta-Zentrale mit dem Fahrrad zurückzulegen. Es war an einem Montag. Ich brauchte keine Steigungen zu bewältigen, aber ein starker Wind blies in die falsche Richtung und machte die Fahrt mühseliger als mir lieb war. Als ich zehn Stunden später nach Hause fahren wollte, hatte der Wind gedreht und blies wieder in die falsche Richtung. Ich rief Tove an, und sie holte mich ab. Es versteht sich von selbst, dass ich nie wieder mit dem Rad zur Arbeit fuhr.

Ich erzähle dieses harmlose Detail, um zu zeigen, dass die Entwicklungen um Linux mein Alltagsleben nicht veränderten. Ein Großteil der damit verbundenen Aktivitäten fand ohnehin in den Unternehmen statt. Technisch orientierte Leute in den Firmen, denen Linux seit langem ein Begriff war, wurden von ihren Chefs angesprochen, die Artikel in Fachzeitschriften gesehen oder auf andere Weise von Linux gehört hatten. Sie wollten von ihren Technikern wissen, was hinter der Euphorie steckte, erkannten die Vorzüge und trafen die Entscheidung, ihre Server unter Linux zu betreiben.

Szenen dieser Art spielten sich in IT Abteilungen auf der ganzen Welt ab, vorwiegend jedoch in den USA. Die Entscheidung basierte nur selten darauf, dass Linux umsonst zu haben war, denn die Anschaffungskosten machen nur einen kleinen Teil einer Software-Investition aus. Service und Support sind weitaus kostspieliger. Was die Anzugträger überzeugte, waren meistens die einfachen technischen Argumente: Linux war stärker als die Konkurrenz - Windows NT und die verschiedenen Unix-Derivate. Dazu kam, dass die Leute es hassen, so handeln zu müssen, wie Microsoft oder ein anderer Anbieter es ihnen vorschreibt. Mit Linux sind Dinge möglich, die bei den Wettbewerbern nicht möglich sind. Die ersten Linux-Anwender setzten Linux ein, weil sie damit in einer Art und Weise auf den Quellcode zugreifen konnten, wie es bei kommerzieller Software nicht möglich war.

So betrachtet hatte sich nichts geändert, seit ich Version 0.01 von meinem Zimmer aus freigegeben hatte. Linux war flexibler als andere Systeme auf dem Markt. Man war damit sein eigener Herr. Und es war, jedenfalls was Webserver anbetraf, nicht so aufgebläht und mit so vielen unnötigen Eigenschaften ausgestattet wie konkurrierende Betriebssysteme.

Und noch etwas sprach für Linux: Trotz seiner wachsenden Popularität als Betriebssystem für Webserver besetzte es keine Nische. Das ist wichtig, wenn man den Erfolg von Linux verstehen will.

Mainframe-Computer waren Nischenprodukte. Unix im Allgemeinen war in den verschiedensten Nischen zu finden - in der Supercomputer-nische des Verteidigungsministeriums, in der Bankennische. Die Leute, die Betriebssysteme für Mainframes und andere große Systeme verkauften, verdienten Geld, weil sie ihre Betriebssysteme zu hohen Preisen absetzen konnten. Dann kam Microsoft und verlangte schlappe neunzig Dollar. Microsoft hatte es nicht auf die Bankennische oder

eine andere Nische abgesehen, sondern war plötzlich überall zu finden. Es war, als würden die Heuschrecken über einen herfallen. Es ist schwer, sich einer solchen Invasion zu entziehen. (Nicht, dass ich etwas gegen Heuschrecken einzuwenden hätte. Ich mag alle Tiere.)

Es ist viel besser, überall präsent zu sein und jede Nische zu besetzen, und genau das war Microsoft gelungen. Stellen Sie sich einen flüssigen Organismus vor, der in jede Ritze eindringt, die er finden kann. Wenn er eine Nische verliert, ist das nicht weiter schlimm. Er breitet sich auf der ganzen Welt aus und sickert in jeden Spalt, der sich vor ihm auftut.

Mit Linux ist es heute genauso. Es sickert in jeden Spalt, der sich auftut. Linux ist nicht auf eine Nische beschränkt. Es ist klein und flexibel und bahnt sich seinen Weg. Sie finden es in Supercomputern, in amerikanischen Großforschungseinrichtungen wie den Fermilabs oder bei der NASA. Wobei diese Anwendungen dem Server-Bereich entsprungen sind. Der seinerseits dem Desktop-Bereich entsprungen ist -

der Punkt, an dem ich angefangen habe. Und gleichzeitig finden Sie Linux in eingebetteten Systemen vom Antiblockiersystem bis zur Armbanduhr.

Beobachten Sie, wie es sich ausbreitet.

Es hat einen großen Vorteil, an der Basis zu sein. Die besten und hellsten Köpfe der nächsten Generation setzen dein Produkt ein, weil sich die Phantasie dieser Generation an dir entzündet. In einer früheren Generation entzündete sich die Phantasie der Leute weniger an Microsoft und DOS als an PCs. Wenn du mit PCs zu tun hattest, hattest du mit DOS zu tun. Du hattest keine große Wahl.

Und das war ein riesiger Vorteil für die Ausbreitung von Microsoft.

Wenn Sie sich die hellsten der Kids von heute ansehen, arbeiten viele, wenn auch nicht alle, mit Linux. Klar, einer der Gründe, warum sowohl die Open-Source-Philosophie als auch Linux an den Hochschulen so viele Anhänger haben, liegt auf der Hand: der Protest gegen das Establishment. (Der gleiche Protest gegen das Establishment, der das Leben meines Vaters so stark beeinflusste.) Für sie geht es um den Kampf zwischen der großen, üblen Microsoft Corporation & dem bösen, gierigen, milliardenschweren Bill Gates gegen die Unser-Idealist-die-Liebe-und-freie-Software-für-alle-Bewegung & den (scheinbar) edelmütigen Volkshelden Linus B. Torvalds. Nach dem Studium treten diese Kids ins Berufsleben ein und tragen ihre Liebe zu Linux in die Unternehmen.

Und so behaupten Leute, die sich in die Tiefen von Microsoft vorgewagt haben, sie hätten mein Gesicht als Zielscheibe auf Dartboards gesehen. Ich sage dazu nur eines: Wie schaffen sie es bloß, meine Nase zu verfehlen?

Aber ich greife mir wieder vor. Auf IBMs große Ankündigung im Frühjahr 1998 folgten ähnliche Ankündigungen jedes großen Hardwareherstellers. Im August hatte die Zeitschrift *Forbes* unsere kleine Welt „entdeckt“ und ein Titelbild mit mir gebracht, das mit „Friede, Liebe, Software“ überschrieben war. Als Unternehmen um Unternehmen sein (unvermeidliches) Engagement für Linux bekundete, musstest du nicht mehr die Advocacy-Newsgroups durchforsten, um den Teesatz zu lesen.

VI.

Linux hatte das Herz des Planeten erobert wie ein Außenseiter aus einem obskuren Dritte-Welt-Land olympisches Gold.

Ich war das Aushängeschild. In einem Presseinterview erklärte Eric Raymond, ein Teil meiner Anziehungskraft (oder so ähnlich) bestünde darin, dass ich weniger gewöhnungsbedürftig aussähe als viele andere Hacker. Okay. Das ist die Meinung eines einzelnen Hackers. Nicht jeder war von dieser Situation begeistert. Richard Stallman setzte sich dafür ein, den Namen Linux in gnu/Linux abzuändern, weil ich, so seine Logik, den GNU-gcc-Compiler und ein paar andere frei zugängliche Softwarewerkzeuge und Anwendungen verwendet hatte, um Linux zu verwirklichen. Er und einige andere reagierten mit zunehmender Irritation auf die Tatsache, dass Linux sich auch in der Industrie zu etablieren begann. Die Presse spielte den Widerspruch zwischen den Idealisten und den Pragmatikern (die Begriffe stammen nicht von mir!) unter den mittlerweile hunderttausenden von Linux-Weggenossen hoch. Nach dieser Einteilung wurden diejenigen, die die Unvereinbarkeit der Linux-Ideale mit den Zielen des Kapitalismus fürchteten, als die Idealisten bezeichnet. Ich war der Anführer der Pragmatiker. Aber ich sah diese Analyse als journalistischen Unsinn an - als vereinfachenden Versuch, alles säuberlich in eine Schwarz-Weiß-Welt einzuordnen. (Ich habe das gleiche Problem mit der Art, wie die Leute das Linux-Phänomen als einen Krieg zwischen Linux und Microsoft begreifen, wo doch eigentlich etwas ganz anderes und viel weitreichenderes dahinter steckt. Es ist eine lebendigere Möglichkeit, Technologie, Wissen, Wohlstand und Spaß zu verteilen als die Welt des Kommerzes sie je gekannt hat.)

Für mich war das alles kein Thema. Wie sollte Linux sich neue Märkte erobern, wenn kein kommerzielles Interesse daran bestand? Wie sonst sollte es Chancen für Innovationen eröffnen? Wie sonst sollte es die Leute erreichen können, die eine Alternative - eine frei zugängliche Alternative - zu der vorhandenen schlechten Technologie anstreben? Welcher realistischere Weg stünde der Open-Source-Bewegung offen, als durch das Sponsoring der Wirtschaft Fuß zu fassen? Und wo, wenn nicht in den Unternehmen, sollte ein Teil der weniger interessanten Arbeit wie Wartung und Support erledigt werden?

Open Source lässt alle mitspielen. Warum sollte die Wirtschaft, die für einen so großen Teil des technologischen Fortschritts verantwortlich ist, davon ausgeschlossen sein - vorausgesetzt, sie hält die Spielregeln ein? Open Source trägt lediglich dazu bei, die von der Industrie entwickelten Technologien noch besser zu machen, und hält eventuell die Gier der Unternehmen in Schach.

Und selbst wenn wir den Kräften des Kommerzialisierung Einhalt gebieten wollten, was können wir letztlich schon tun? Ich war nicht bereit, vorzuschlagen, dass wir uns verstecken, uns in den Untergrund zurückziehen und jedes Gespräch mit Leuten aus der Wirtschaft vermeiden sollten.

Die Open-Source-Gemeinde war immer auch von anti-kommerziellen Einstellungen geprägt, aber das wurde erst zum Problem, als Linux auch wenig technisch interessierten Leuten ein Begriff zu werden begann. Das paranoide Geschrei von ein paar lautstarken Verrückten brachte die Newsgroups auf die Barrikaden. Keiner der Linux-Entwickler, mit denen ich zusammenarbeitete, war auch nur im Geringsten beunruhigt. Aber andere ereiferten sich in heiligem Zorn, Red Hat oder andere Unternehmen pervertierten die Gedanken der OpenSource-Bewegung und manche Leute seien dabei, ihren Idealismus zu verlieren.

Bis zu einem gewissen Grad ist es wahrscheinlich richtig, dass einige Open-Source-Leute tatsächlich Gefahr liefen, vom Pfad des Idealismus abzukommen. Aber während bestimmte Leute diese Aussicht als Verlust ansahen, empfand ich sie einfach als Erweiterung unserer Möglichkeiten. Leute mit einem technischen Hintergrund, die sich zum Beispiel darüber Sorgen machten, wie sie ihre Kinder ernähren sollten, hatten nun eine Option. Du konntest weiterhin deinem Idealismus frönen oder du konntest dich dafür entscheiden, in die neue kommerzielle

Szene einzusteigen. Du verlierst nichts, wenn ein anderer hereinschneit und dir eine neue Option eröffnet. Vorher blieb dir nichts anderes übrig, als auf dem Pfad der Tugend zu wandeln.

Ich hatte übrigens nie das Gefühl, im idealistischen Lager zu stehen. Natürlich habe ich Open Source immer als eine Möglichkeit gesehen, die Welt zu verbessern. Vor allem aber sehe ich es als Möglichkeit, Spaß zu haben. Das ist nicht besonders idealistisch gedacht.

Und ich habe schon immer gefunden, dass idealistische Leute interessant, aber irgendwie langweilig und manchmal beklemmend sind.

Um eine sehr starke Meinung zu vertreten, musst du alle anderen Meinungen ausschließen. Dafür bedarf es einer gewissen Verbohrtheit -einer Verbohrtheit, die mir zum Beispiel auch an der amerikanischen Politik im Vergleich zur europäischen missfällt. In der amerikanischen Version des Spiels ziehst du die feindlichen Linien, und der Entschlossenere gewinnt. Europäische Politiker dagegen gewinnen eher, indem sie Kooperationsbereitschaft demonstrieren.

So bleibt mir nur der versöhnliche Ansatz. Der Gedanke an eine kommerzielle Nutzung machte mich nur ein einziges Mal nervös: gleich ganz am Anfang, als Linux noch keinen rechten Namen hatte. Wenn damals kommerzielle Leute die Ziele von Linux verraten hätten, hätte ich nichts dagegen tun können. Heute ist das offensichtlich anders. In den Newsgroup-Diskussionen der Geschehnisse des Jahres 1998 wurde unter anderem die Sorge geäußert, von den Entwicklern in den Unternehmen würde nichts zurückkommen. Bis zu einem gewissen Grad musste ich den neuen Mitspielern aus der Wirtschaft das gleiche Vertrauen entgegenbringen, das die Linux-Entwickler mir entgegengebracht hatten. Und sie erwiesen sich als vertrauenswürdig. Sie haben uns nichts vorenthalten. Bisher verlief die Zusammenarbeit sehr positiv.

Als Aushängeschild, Inhaber des Linux-Warenzeichens und Verantwortlicher für den Linux-Kernel verspürte ich ein wachsendes Gefühl der Verpflichtung. Ich fühlte mich immer stärker für die Tatsache verantwortlich, dass sich mittlerweile Millionen von Menschen auf Linux verließen, und empfand einen ungeheueren Druck, sicherzustellen, dass es so verlässlich wie möglich arbeitete. Es war mir wichtig, der Wirtschaft verständlich zu machen, was hinter dem Open-SourceGedanken stand. Soweit es mich betraf, gab es keinen Krieg zwischen den gierigen Aktiengesellschaften und den altruistischen Hackern.

Nein, ich gab meine Ideale nicht auf, nur weil ich mich mit den Intel-Leuten traf, als sie mich baten, ihnen bei der Behebung des PentiumFO OF-Fehlers zu helfen. („Pentium FO OF-Fehler?“ höre ich Sie fragen. Tja, verschroben wie wir sind, haben wir Ingenieure eine Vorliebe für verschrobene Namen. „FO OF“ ist die hexadezimale Repräsentation der beiden ersten Bytes einer unzulässigen Befehlssequenz, die dazu führte, dass sich der Pentium-Prozessor aufhängte. Daher der Name.) Nein, es war nicht heuchlerisch, die Wunder des Open-Source-Codes in die Welt zu tragen, während ich ein Gehalt von einer Firma bezog, die so verschlossen war, dass sie den Leuten nicht einmal sagte, was sie eigentlich machte. Tatsache ist, ich respektierte und respektiere den von Transmeta entwickelten Chip mit geringer Leistungsaufnahme als das interessanteste und vielversprechendste Technologieprojekt auf dem Markt - und dasjenige mit dem folgenreichsten Potenzial. Und, nur um es ein für alle Mal festzuhalten, ich war an den Bemühungen beteiligt, das Unternehmen dazu zu bewegen, wenigstens einen Teil seines Codes offen zu legen.

Ich verspürte den Druck, in der Open-Source-Gemeinde weiterhin als technisch und ethisch vertrauenswürdig zu gelten. Es war mir wichtig, für keines der konkurrierenden Linux-Unternehmen Partei zu ergreifen. Nein, ich habe mich nicht verkauft, als ich die Aktienoptionen akzeptierte, die Red Hat mir freundlicherweise als Dank anbot. Dagegen erschien es mir richtig, den Londoner Unternehmer abzuweisen, der mir 10 Millionen Dollar anbot, nur damit ich als Board-Mitglied meinen Namen für sein kaum flügge gewordenen Linux-Unternehmen hergab. Er konnte nicht nachvollziehen, dass ich eine derart große Summe für einen so kleinen Gefallen ausschlug. Verständnislos wollte er wissen: „*Was verstehen Sie an zehn Millionen Dollar nicht?*“

Es war mir nicht in den Sinn gekommen, dass ich mit solchen Problemen konfrontiert sein würde. Linux' neu erworbene Popularität brachte ein paar heikle Situationen mit sich, nicht nur für mich, sondern für die ganze virtuelle Gemeinde. Als 1998 alle Welt auf OpenSource-Code aufmerksam wurde, entbrannte unter anderem eine große Debatte um den Namen selbst. Bis dahin hatten wir das Phänomen, Software unter Lizenzen wie der GPL frei zugänglich zu machen, als „freie Software“ bezeichnet und ganz allgemein von der „freien Softwarebewegung“ gesprochen. Der Begriff wurzelte in der Free Software Foundation, die Richard Stallman 1985 gegründet hatte, um

freie Softwareprojekte wie GNU, das von ihm lancierte freie Unix, zu fördern. Plötzlich befanden Missionare wie Eric Raymond, das könnte bei den Journalisten zu Verwirrung führen. Bedeutete das Wort „frei“, dass die Software nichts kostete? Bedeutete es „frei“ im Sinn von „uneingeschränkt“? Bedeutete es „frei“ wie in „Freiheit“? Es stellte sich heraus, dass Brian Behlendorf, der mit Journalisten über Apache sprach, auf ähnliche Sprachprobleme stieß. Nach einem wochenlangen privaten E-Mail-Austausch, an dem ich nicht teilnahm, sondern über den ich lediglich per cc: informiert wurde (ich hatte an der politischen Seite kein Interesse), wurde eine Einigung erzielt: Wir würden uns als „offen“ statt als „frei“ bezeichnen. Deshalb wurde aus der freien Softwarebewegung die offene Softwarebewegung, für Leute, die sie gern als eine Bewegung sehen wollten, was sie ja wohl auch war. Die Free Software Foundation heißt immer noch Free Software Foundation, und Richard Stallman ist weiterhin ihr psychologischer Vordenker.

Als einer der De-facto-Anführer dieser Bewegung war ich zunehmend gefragt. Jedes Mal, wenn mein Telefon bei Transmeta klingelte - und es klingelte in jenen Tagen die ganze Zeit - bedeutete das eines von zwei Dingen: Entweder ein Journalist wollte mich für ein Interview gewinnen oder die Organisatoren einer Tagung für einen Vortrag. Um die Kunde von Open Source und Linux zu verbreiten, fühlte ich mich verpflichtet, beides zu akzeptieren. Nehmen Sie ein schüchternes Mathegenie, schicken Sie es auf Grüß-und-Grins-Tour für eine populistische Sache, und Sie haben einen Volkshelden kreiert. Vergessen Sie Eric Raymonds Spruch, mein Aussehen sei weniger gewöhnungsbedürftig als das vieler anderer Hacker. Ein großer Teil meiner Anziehungskraft (oder so) bestand darin, dass ich *nicht* Bill Gates war.

Journalisten schienen von der Tatsache fasziniert zu sein, dass Gates in einem High-Tech-Schloss am See wohnte, während ich in einer Vierzimmer-Doppelhaushälfte mit veralteter Installation über die Spielsachen meiner Töchter stolperte. Oder einen langweiligen Pontiac fuhr. Oder selbst ans Telefon ging. Wer wäre davon nicht fasziniert?

Als Linux anfing, als echte Bedrohung für Microsoft zu gelten - und in den Zeiten seiner Antitrust-Nöte konnte Microsoft dringend zumindest den Anschein einer echten Bedrohung gebrauchen -, griff die Presse jede Entwicklung so begierig auf, als ginge es um den Dritten Weltkrieg. Jemand ließ das „Halloween Document“ durchsickern, ein

internes Microsoft-Memo, aus dem hervorging, dass sich das Unternehmen wegen Linux Sorgen machte. Bald wurde Steve Balmer mit den Worten zitiert: „Klar bin ich beunruhigt.“ So gut es Microsoft ins Konzept passen mochte, die Konkurrenz künstlich hochzuspielen, die Windows NT durch Linux erfuhr - Tatsache war, dass der Wettbewerb in Zukunft auf jeden Fall heftiger werden würde.

Ich hatte es *nicht nötig*, auf eine Seifenkiste zu steigen und schreckliche Dinge über Microsoft zu sagen. Wozu hätte das gut sein sollen? Die Ereignisse sprachen für sich. Und sie sprachen zugunsten von Linux. Die Journalisten liebten das Spektakel. Der leise (wie ein Fuchs) auftretende David gegen den monopolistischen, gewissenlosen Goliath. Und, unter uns, es gefiel mir, dieses Thema mit Reportern zu bereden. Ich bezeichne Journalisten gern als Abschaum, aber die meisten meiner Interviews haben mir wirklich Spaß gemacht. Die meisten Reporter interessierten sich für unsere Seite der Story - wer schlug sich nicht gern auf die Seite des Underdog?

Wenn sie von der Geschichte der Amöbe, die Microsoft zerstörte, genug hatten, wollten die Journalisten unweigerlich mehr über das Open-Source-Konzept erfahren. Diese Information ließ sich in immer kürzerer Zeit vermitteln, weil die Leute nun an Beispielen sehen konnten, wie die Sache funktionierte. Als Nächstes schienen sie sich über die Verwaltung von Linux zu wundern. Es war ihnen schleierhaft, wie das größte Gemeinschaftsprojekt in der Geschichte der Menschheit so effektiv gemanagt werden konnte, wenn es schon in Durchschnittsfirmen mit 30 Mitarbeitern nicht selten wie in einem Stall aufgescheuchter Hühner zuing.

Jemand prägte den Begriff „Gütiger Diktator“, um zu beschreiben, wie ich die ganze Sache leitete. Als ich diese Bezeichnung zum ersten Mal hörte, kam mir unwillkürlich der General eines Sonnenstaates mit dunklem Schnurrbart in den Sinn, der an seine hungrigen Landeskinder Bananen verteilt. Ich weiß nicht, ob ich mich mit dem Image des „Gütigen Diktators“ wohl fühle. Ich kontrolliere den Linux-Kernel, den Unterbau des Ganzen, weil bisher jeder, der mit Linux zu tun hat, mir mehr als jedem anderen vertraut. Bis heute manage ich das Projekt mit Hunderttausenden von Entwicklern nach der gleichen Methode wie damals, als ich noch in meinem Zimmer saß und wie wild programmierte: Statt Arbeit aktiv zu delegieren, warte ich eher darauf, dass die Leute auf mich zukommen und sich der Sache freiwillig

annehmen. Das begann schon damals, als ich mich von Aufgaben wie der Programmierung der Bedieneroberfläche befreite, die ich weniger interessant fand. Leute kamen auf mich zu und boten an, einzelne Teilsysteme zu übernehmen. Und über die Leute, die für die Wartung dieser Teilsysteme verantwortlich sind, bekomme ich alles mit, was passiert.

Ich nehme ihre Arbeit an oder lehne sie ab, aber meistens lasse ich den Dingen ihren Lauf. Wenn zwei Leute ähnliche Programme pflegen, akzeptierte ich beide und warte ab, welches genutzt wird. Manchmal werden beide genutzt, schlagen aber zu guter Letzt unterschiedliche Wege ein. Einmal gab es einen heftigen Wettbewerb zwischen zwei Leuten, die darauf bestanden, Patches einzusenden, die jeweils im Widerspruch zu denen des anderen standen. Ich nahm daraufhin von keinem der beiden Entwickler mehr Patches an, bis einer von ihnen das Interesse verlor. Hätte König Salomon einen Kindergarten geleitet, hätte er genauso gehandelt.

Gütiger Diktator? Nein, ich bin einfach nur faul. Ich versuche zu managen, indem ich keine Entscheidungen treffe und den Dingen ihren Lauf lasse. So bekommst du die besten Ergebnisse.

Das machte Schlagzeilen.

Die Ironie dabei ist, dass mir mein Management-Stil, wenn man ihn als solchen bezeichnen kann, zwar gute Noten bei der Presse einbrachte, dass ich aber während meines kurzen Gastspiels als Manager bei Transmeta definitiv versagte. Damals wurde beschlossen, dass ich ein Entwicklerteam leiten sollte. Ich floppte. Wie jeder, der je in meine Müllhalde von Büro vorgedrungen ist, weiß, bin ich völlig unorganisiert. Ich hatte Probleme, die wöchentlichen Projektstatusbesprechungen, die Leistungsüberprüfungen, die Liste der offenen Punkte zu managen. Nach drei Monaten war für jedermann sichtbar, dass mein Managementstil nicht dazu beitrug, Transmeta voranzubringen, trotz der Anerkennung, die ich von der Presse für meine Art bekam, Linux zu leiten.

In der Zwischenzeit nahm die Presse ein anderes Thema unter Dauerbeschuss: Zersplitterung. Wer die wüste, unglückliche Geschichte von Unix verfolgt hat, kennt die endlosen Reibereien zwischen den Unix-Anbietern. 1998 wurde andauernd die Frage gestellt: Wird sich die Geschichte in der Linux-Welt wiederholen? Meine Antwort lautete immer, es gäbe zwar zweifellos Reibereien zwischen Linux-Anbietern,

sie könnten jedoch nicht zu der Zersplitterung führen, die Unix zum ewigen Knapp-daneben-ist-auch-vorbei gemacht hat. Das Problem bei Unix war, dass konkurrierende Anbieter Jahre darauf verschwendet hatten, ähnliche Eigenschaften zu implementieren, einfach weil sie nicht auf eine gemeinsame Codebasis zugreifen konnten. Die unabhängige Entwicklung der immer gleichen Eigenschaften kostete Unix nicht nur Jahre, sie führte auch zu blutigen internen Kämpfen. Klar, erzählte ich der Presse, auch Linux-Anbieter treffen sich nicht zu regelmäßig stattfindenden Love-ins. Aber es wird in der Linux-Gemeinde weniger Zersplitterung geben, weil sogar Linux-Anbieter, die sich nicht leiden mögen, auf die gleiche Codebasis zugreifen und die Arbeit des jeweils anderen nutzen können. Der Quellcode ist ein Topf, aus dem sich jeder bedienen kann.

Je besser Journalisten solche Konzepte zu verstehen begannen, desto lieber traf ich mich mit ihnen. (Anders als die Journalisten, die ich in meiner Jugend in Helsinki kennen gelernt hatte, waren die meisten amerikanischen Journalisten der 1990er-Jahre nüchtern.) Ich nahm gerne die Gelegenheit wahr, mit ihnen zu diskutieren.

Aber Vorträge waren etwas ganz anderes. Ich bin niemand, der sich gerne produziert. Sie erinnern sich: Ich bin der Typ, der in seiner Kindheit kaum jemals aus seinem Zimmer herauskam. Ich war noch nicht einmal gut darin, Vorträge zu *schreiben*, und schob die Vortragsvorbereitung grundsätzlich bis zum Vorabend vor mir her.

Irgendwie schien das keine Rolle zu spielen. Meistens trat ich ans Rednerpult und die Leute erhoben sich und applaudierten, bevor ich überhaupt den Mund aufgemacht hatte. Ich möchte nicht undankbar klingen, aber ich habe das immer als peinlich empfunden. Alles, was du in dieser Situation sagst, klingt falsch, sogar mein Standardsatz: „Danke, nehmen Sie doch bitte Platz.“ Ich nehme Alternativvorschläge gerne entgegen.

Okay, nicht alle Anrufe kamen von Journalisten oder Tagungsorganisatoren. Eines Abends, als Tove und ich den Mädchen vorlasen, klingelte das Telefon.

Ich hob ab: „Torvalds.“

„Ähm. Ist das der Linux-Typ?“

„Ja.“

Zwei Sekunden Stille. *Klick*.

An einem anderen Abend rief mich ein Typ aus Las Vegas zu Hause an und wollte mich dazu bringen, in irgendein Geschäft mit Linux-T-Shirts einzusteigen.

Am naheliegendsten wäre es gewesen, eine Geheimnummer zu beantragen. Als wir nach Kalifornien zogen, habe ich mich nicht darum bemüht, weil sie teurer als eine normale Nummer gewesen wäre. Seither habe ich gelernt, welchen Preis du für so viel Sparsamkeit zahlst, und jetzt habe ich eine Geheimnummer. Einmal, noch bevor ich sie beantragt hatte, hatte David meine Privatnummer verlegt und die Telefonauskunft angerufen. Er fragte den Typen nach meiner Nummer. Der Typ gab sie ihm und sagte höchst erstaunt: „*Er steht im Telefonbuch? Mit all seinen Millionen?*“

Aber es gab keine Millionen. Millionen von Linux-Anwendern, klar. Aber keine Dollar-Millionen für Linus.

Und das war völlig okay.

VII.

An den meisten Tagen wache ich auf und denke, ich bin der größte Glückspilz der Welt. Ich weiß nicht mehr, ob Mittwoch, der 11. August 1999 so ein Tag war, er hätte es jedenfalls sein müssen.

Es war der zweite Tag der Linux World, die im Tagungszentrum von San Jose stattfand. Dirk Hohndel, der Geschäftsführer von SuSE, der aus Deutschland zur Messe angereist war, hatte die Nacht auf dem Gästebett in unserem Wohnzimmer verbracht. Ich kenne ihn seit Jahren. Er ist einer der alten XFree86-Leute, und befasst sich mit der Grafikseite von Linux. Außerdem ist er Danielas Patenonkel. Ich wachte auf, machte für Tove und Dirk Cappuccino, las wie immer die San Jose Mercury News von der ersten bis zur letzten Seite, außer dem Sportteil und den Stellenanzeigen, und dann zwängten wir uns in den Toyota Rav4 und fuhren die zehn Meilen ins Stadtzentrum nach San Jose hinunter.

Ich weiß noch, dass ich eine Menge Hände schütteln musste.

An diesem Tag sollte Red Hat an die Börse gehen. Das Unternehmen hatte mir vor Jahren Aktienoptionen gegeben, und mir erst kurz zuvor ein paar Unterlagen geschickt, die ich mir aber bis dahin noch nicht einmal angesehen hatte. Sie waren irgendwo in einem Stapel Papier neben meinem Computer vergraben. Ich erinnere mich daran, dass ich Red Hat wirklich die Daumen drückte. Das lag weniger an meiner Aufregung wegen der Aktienoptionen - mir war nicht ganz klar, was sie bedeuteten. Meine Fiebrigkeit hatte einen anderen Grund. In vieler Hinsicht würde ein erfolgreicher Börsengang eine Bestätigung von Linux sein. Aus diesem Grund war ich an diesem Vormittag nervös, aber das ging nicht nur mir so. Die Stimmung am Markt war seit Wochen schlecht gewesen, und die Leute fragten

sich, ob der Börsengang von Red Hat überhaupt ein Erfolg werden würde.

Aber der „Liquiditätsschub“ trat tatsächlich ein. In den Konferenzsälen erfuhren wir, dass der Ausgabekurs von Red Hat auf 15 Dollar festgelegt worden war. Oder waren es 18 Dollar gewesen? Ich weiß es nicht mehr. Wichtig war nur, dass die Aktie am Ende des ersten Handelstages mit 35 Dollar notiert wurde. Das war nicht rekordverdächtig, aber okay.

Ich erinnere mich, wie ich mit Tove und Dirk im Auto nach Hause fuhr und zuerst mal erleichtert war. Dann dachte ich an das Geld und spürte, wie die Aufregung in mir hochstieg. Erst als wir in den Highway 101 Richtung Norden einbogen, realisierte ich, dass mein Vermögen an einem einzigen Tag von praktisch null auf eine halbe Million Dollar angewachsen war. Mein Herz fing an, schneller zu schlagen. Ich war auf Wolke sieben und konnte es nicht glauben.

Ich hatte keine Ahnung von Aktien und beschloss, herauszufinden, was als Nächstes zu tun war. Deshalb rief ich Larry Augustin an, den CEO von VA Linux. Ich sagte ihm, er sei der einzige Mensch, den ich kannte, der eine Ahnung von Aktien hatte. Wortwörtlich sagte ich: „Hast du so etwas wie einen Börsenmakler oder jemanden, dem du vertraust? Ich will damit nicht zu eBay gehen.“

Red Hat hatte mir Optionen gegeben - kein richtiges Aktienpaket. Ich wusste nicht, wie ich sie ausüben konnte. Ich wusste, dass es möglicherweise eine Sperrfrist gab, aber mit war nicht bekannt, ob sie auch für mich galt, und ich hatte nicht an die steuerlichen Folgen gedacht. Larry, der gut in diesen Dingen ist und eine Menge Leute kennt, stellte einen Kontakt zu einem Typen bei Lehman Brothers her, der für kleine Fische wie mich eigentlich gar nicht zuständig war. Er versprach mir, herauszufinden, was ich tun sollte. Inzwischen hatte ich, zwei Tage nach dem Börsengang, eine E-Mail von jemandem aus der Personalabteilung von Red Hat oder einem Anwalt bekommen, aus der hervorging, dass die Aktien vor dem Börsengang gesplittet worden waren. Das sagte mir überhaupt nichts. Deshalb spürte ich die Infomappe auf, die zu lesen ich mir bisher nicht die Mühe gemacht hatte, und dort stand in (für Juristenverhältnisse) einfachem Englisch: Meine Optionen hatten sich wundersamerweise verdoppelt.

Meine halbe Million Dollar war eigentlich *eine Million Dollar*.

Ganz ehrlich: Auch wenn die Presse mich ständig als uneigennützigem, einem Armutsgeplöde verpflichteten Geek-für-die-Massen hinstellt, drehte ich vor Freude glatt durch.

So, jetzt wissen Sie es.

Ich setzte mich hin und las die Red-Hat-Unterlagen von vorne bis hinten durch. Ja, ich musste eine Sperrfrist von 180 Tagen einhalten.

Können Sie ermessen, wie lang 180 Tage sein können, wenn man zum ersten Mal in seinem Leben auf dem Papier Millionär geworden ist?

Von da an hatte ich einen neuen Sport (oder einen Sport, Punkt): den Kurs der Red-Hat-Aktie zu verfolgen, die ihren Höhenflug in den folgenden sechs Monaten fortsetzte. Unterstützt von ein paar gewaltigen Sprüngen nach oben stieg ihr Kurs kontinuierlich höher und höher. Irgendwann wurde die Aktie noch einmal gesplittet. An meinem besten Tag war ich dank der Optionen 5 Millionen Dollar wert.

Red Hat fing relativ niedrig an und rückte dann, als die Wall Street in ihrem Liebeswahn zu allem, was nur im Entferntesten mit dem Internet zu tun hatte, Linux „entdeckte“, dem Himmel Stück für Stück näher. Wir waren in den kalten Wintermonaten Ende 1999 das Eis des Monats. Investmentexperten in Fernsehen und Presse konnten nicht genug von diesem verrückten kleinen Betriebssystem bekommen, das versprach, Microsoft aus dem Konzept zu bringen. Bei mir klingelte pausenlos das Telefon. Das Ganze erreichte seinen unglaublichen Höhepunkt mit dem Börsengang von VA Linux am 9. Dezember. Das Zeichnungsinteresse übertraf alle Erwartungen.

Larry Augustin und ich fuhren hoch nach San Francisco, um den Börsengang in den Geschäftsräumen der First Boston Credit Suisse vor Ort zu verfolgen. Ich trug, was ich immer trage: Ein T-Shirt mit Werbeaufdruck und Sandalen. Wir hatten unsere Frauen und Kinder dabei. Es war eine verrückte Szene, wie die Kleinkinder zwischen den zugeknöpften Investment-Bankern herumtobten.

Dann ging alles Schlag auf Schlag. Auf den Monitoren zogen Zahlen vorbei, die zeigten, dass VA Linux am ersten Handelstag um die 300 Dollar pro Aktie kostete. Das hatte es noch nie gegeben. Selbst wenn wir die Zahlen nicht gesehen hätten - an der Art, wie die Investmentbanker hypnotisiert auf CNN und Bloomberg starrten, hätten wir gemerkt, dass das ein Rekordergebnis war. Larry war sein übliches cooles Selbst. Ich glaube, er zuckte während der ganzen Sache nicht

einmal mit der Wimper. Aber genau weiß ich es nicht, weil ich damit beschäftigt war, meinen Töchtern hinterher zu jagen.

Wahrscheinlich haben sogar die Bewohner der Regenwälder in Madagaskar mitbekommen, wie reich Larry durch die Sache wurde. Er war als finanzielles Leichtgewicht nach San Francisco hochgefahren und bei der Rückkehr ins Silicon Valley um die 1,6 Milliarden Dollar schwer. Und dabei war er, wie die Presse nicht müde wurde zu erwähnen, noch nicht einmal dreißig.

Ich für meinen Teil hatte ein Aktienpaket und Aktienoptionen erhalten. Wie bei Red Hat konnte ich die Aktien sechs Monate lang nicht verkaufen. Aber im Gegensatz zu Red Hat, das kontinuierlich stieg, kannte VA Linux nur eine Richtung: nach unten. Nach dem alle Rekorde brechenden Eröffnungstag fiel der Kurs ein Jahr lang und erreichte einen Tiefststand von 6,62 Dollar. Teilweise war die Aktie zum Opfer der Marktkorrektur geworden, die die meisten Aktien im April gebeutelt hatte. Aber darüber hinaus endete mit dem frühlingshaften Tauwetter auch Linux' Gastspiel als Eis des Monats. Wegen der VA Linux-Sperrfrist konnte ich den Höhenflug des Marktes nicht nutzen. Psychologisch war es viel schwieriger, den Kurs von VA Linux zu verfolgen als den von Red Hat: jeden Abend mit dem Wissen ins Bett zu kriechen, dass ich morgens beim Aufwachen (finanziell) weniger wert sein würde.

Trotzdem fühle ich mich nach wie vor als der größte Glückspilz der Welt.

An einem Abend im Januar besucht Linus mich in meinem Büro in Sausalito. Nachdem er sich über meinen Mac und mein Nicht-Linux-Betriebssystem lustig gemacht hat, setzt er sich hin und liest den ersten Entwurf des langen Vorworts, das ich in der ersten Person, aus seiner Perspektive, geschrieben habe. Ich sitze vielleicht fünf Zentimeter neben ihm. Von Linus ist nur einmal etwas zu hören: als er über eine Zeile stolpert, in der steht, dass er nie erwartet hätte, der einzige Weltstar neben Jean Sibelius und Nikki, dem Rentier, zu werden, den Finnland je hervorgebracht hat. Nach vielleicht zehn Minuten hat er das Vorwort durch, und sein einziger Kommentar lautet: „Mann, schreibst du lange Sätze.“ Wir verbringen ein paar Stunden damit, die Sätze zu kürzen, Wörter einzubauen, die er wirklich verwenden würde, und zu lernen, wie wir zusammen arbeiten können, nachdem wir uns bereits bewiesen haben, dass wir es meisterhaft beherrschen, gemeinsam abzuhängen. Schließlich warfen wir das Vorwort in den Papierkorb.

Dann versucht Linus erfolglos, die Auflösung meines Flachbildschirms zu verbessern. Er war im Vorjahr brandneu und für mich bis dahin eine Art Statussymbol gewesen. „Wie kannst du von diesem Ding bloß lesen?“, fragt er. Er schafft es nicht, die Auflösungsqualität so zu verbessern, dass sie seinen Vorstellungen entspricht. Dann nimmt er ein Blatt Papier und fängt an, Diagramme zu zeichnen und zu erklären, wie Bildschirme funktionieren. Irgendwann sage ich: „Hey, gehen wir Sushi essen.“

„Diese Geldangelegenheiten machen mich verrückt. Darauf zu warten, dass die Sperrfrist abläuft. Ich fühle mich, als hätte ich einen Haufen Geld, aber dann doch wieder nicht. Es lässt mich nicht los.“

Ich bestelle Sake. Er bestellt Saft, weil er fährt.

„Bis jetzt hatten wir fast nie mehr als 5000 Dollar auf unserem Girokonto. Außer ein paar Aktien und Ersparnissen, an die wir nicht rankommen, war das alles, was wir ausgeben konnten. Und jetzt habe ich auf dem Papier so viel Geld und...“

„Wie viel Geld? Ein paar Millionen?“

„So um die 20 Millionen Dollar. So viel sind die Aktien aus dem Börsengang von VA Linux wert, solange der Kurs nicht sinkt. Aber ich komme an das Geld erst nach der Sperrfrist von sechs Monaten ran. Nein, jetzt sind es nur noch fünf Monate.“

„Ich sehe da kein Problem. Du musst fünf Monate warten, bevor du dir ein großes Haus kaufen kannst? Ich will ja nicht herzlos klingen, aber...“

„Hey, auf den ersten Blick sieht es so aus, als hätten wir genug Geld, um uns jedes Haus leisten zu können, das uns gefällt. Aber wir brauchen ein Haus mit fünf Schlafzimmern und wir wollen ein Grundstück, das groß genug ist, dass wir die Tiere hören können, und ich habe im Büro jeden Tag Pool-Billard gespielt und deshalb brauchen wir einen Raum, in den ein Billardtisch passt. Und wir hätten gern einen separaten Bereich, wenn Toves Eltern zu Besuch kommen oder damit Freunde meiner Schwester für ein paar Monate aus Finnland herüber kommen und uns bei der Betreuung der Kinder helfen können. Es ist komisch, als wir Patricia bekamen, waren wir gerade dabei, von Finnland in die USA umzuziehen. Als Daniela auf die Welt kam, sind wir aus dem Apartment in die Doppelhaushälfte umzogen und...“

„Ihr arbeitet also daran, noch ein Kind zu bekommen?“

„Na ja, wir lassen den Dingen einfach ihren Lauf“

„Dort, wo ich herkomme, heißt das ‚Wir versuchen, noch ein Baby zu bekommen‘, du Blödmann.“

„Okay. Und deshalb brauchen wir mehr Platz und wir haben uns ein paar Häuser angeschaut und die Häuser, die es gibt, sind unglaublich teuer. Ich meine, da hast du 20 Millionen Dollar und denkst, wow, ich kann mir jedes Haus der Welt leisten. Aber wir haben uns ein Haus für 1,2 Millionen Dollar in Woodside angesehen, das nicht einmal einen Garten hatte und wirklich heruntergekommen war. Das beste Haus, das wir gesehen haben, kostet 5 Millionen Dollar. Aber wenn du 20 Millionen Dollar hast, musst du dir klar machen, dass die Hälfte davon für Steuern draufgeht. Dann bleiben dir noch 10 Millionen Dollar, mit denen du rechnen kannst, aber die Steuern für ein Haus wie dieses könnten im Jahr 60000 Dollar oder so verschlingen, deshalb musst du Geld dafür zurücklegen. Und überhaupt, ich weiß

nicht. Das ist das einzige Mal in meinem Leben, wo ich so viel Geld haben werde, und ich will mich nicht übernehmen und Gefahr laufen, mir nachher das Leben in dem Haus nicht leisten zu können. Und wir wollen uns keine Hypothek aufhalsen. "

„Du kannst mir wirklich nicht Leid tun. Erstens sahnst du tüchtig ab, wenn Transmeta bei seinem Börsengang tüchtig absahnt. "

„Ja, schon. Aber ich bin nur Junior Engineer, deshalb bekomme ich nicht so viele Aktien. Und mein Gehalt ist auch nicht so besonders hoch. "

„Linus, du könntest zu jedem Venture-Kapitalgeber dieser Stadt gehen und so viel kriegen, wie du nur wolltest . .. "

„Da magst du schon Recht haben. "

Villa

Jetzt ist der Moment gekommen, meine goldenen Regeln anzubringen. Die erste lautet: „Verhalte dich anderen gegenüber so, wie du möchtest, dass sie **sich dir gegenüber verhalten.**“ Wenn du dieser Regel folgst, weißt du in jeder Situation, wie du dich zu verhalten hast. Die zweite lautet: „Sei stolz auf das, was du tust.“ Und die dritte: „Und hab deinen Spaß dabei.“

Natürlich ist es nicht immer einfach, stolz auf sich zu sein und dabei auch noch Spaß zu haben. Einen Monat vor dem Börsengang von VA **Linux**, bei einem Einführungsvortrag auf der Comdex **1999** in Las Vegas, gelang mir weder das eine noch das andere. Die Comdex ist, wie jeder weiß, die größte **und schlechteste Computermesse, die die Menschheit** je gesehen hat. Fast eine Woche lang übt die verschlafene Stadt Las Vegas in Nevada eine geradezu **magnetische Anziehungskraft auf alle nur denkbaren Hightech-Produkte, die sich möglicherweise verscherbeln lassen, und die Massen von Leuten aus, die sie kaufen und verkaufen.** Es ist das einzige Mal im Jahr, wo du das Fenster eines Taxis herunterlassen, dich hinausbeugen und eine vorbeistolzierende Stricherin fragen kannst: „Um wie viel Uhr findet die **Keynote-Ansprache** statt?“, und sie die Antwort weiß.

Es war ein Riesending, dass die Messeorganisatoren den gütigen Diktator des Planeten Linux baten, eine **Keynote-Ansprache** bei der Comdex zu halten. Die Computerbranche erkannte damit an, dass Linux eine Kraft war, mit der man rechnen musste.

Bill Gates hielt seine **Keynote-Rede** am Sonntag, dem ersten Messeabend. Er hatte ein Publikum angezogen, das stehend den Ballsaal des Venetian Hotels füllte, der ungefähr so groß ist wie sieben **IKEA-Möbelhäuser** zusammen. Konferenzteilnehmer, die darauf brannten,

Gates' Ausführungen über das damals laufende Antitrust-Verfahren zu hören, oder die einfach eines Tages ihren Enkelkindern erzählen wollten, sie hätten den reichsten Mann der Welt leibhaftig gesehen, standen Stunden vorher im Kellergeschoß des Konferenzzentrums des Hotels Schlange. Gates' Rede begann mit einem Anwaltswitz und umfasste eindrucksvoll choreographierte Vorführungen der InternetTechnologie von Microsoft und hoch professionelle Videosequenzen, die das Publikum in Lachsalven ausbrechen ließen - in einer davon war Gates zu sehen, wie er Austin Powers imitierte. Ich war nicht dabei. Ich half Tove, einen Badeanzug zu kaufen.

Am folgenden Abend hielt ich meinen Vortrag im gleichen Saal.

Ich wäre lieber einkaufen gegangen. Na ja, eigentlich auch nicht ...

Es war nicht so, dass ich nicht vorbereitet gewesen wäre. Normalerweise schreibe ich meine Rede am Vortag, aber dieses eine Mal hatte ich tatsächlich einen Vorsprung. Die Rede fand am Montagabend statt; und ich hatte sie am Samstag geschrieben und den Computer vorbereitet, der die Folien projizieren sollte. Die Sache ließ sich bestens an. Ich hatte meinen Vortrag auf drei verschiedenen Disketten abgespeichert, um auf der sicheren Seite zu sein, falls sich eine von ihnen als fehlerhaft erweisen sollte. Wenn ich etwas mehr hasse, als einen Vortrag zu halten, so ist es, einen Vortrag zu halten, bei dem etwas schief geht. Ich stellte meine Rede sogar ins Internet, falls sich *alle* Disketten als fehlerhaft erweisen sollten.

Auf dem Strip gerieten wir in einen Comdex-bedingten Stau, so dass wir erst eine halbe Stunde vor Beginn des Vortrags im Venetian Hotel eintrafen. Ich hatte Tove und die Mädchen und ein paar Leute von der Messe dabei. Als ich endlich das Gebäude betrat, hatten wir Probleme, uns Zutritt zu den Räumen hinter der Bühne zu verschaffen, weil einer der Organisatoren unsere Anstecker verlegt hatte. Kurz gesagt, alles was schief gehen konnte, ging schief.

Endlich kamen wir hinein. Ich wäre schon nervös gewesen, wenn ich vor 40 Leuten hätte sprechen müssen, geschweige denn vor dem größten Publikum meines Lebens. Und dann passierte es.

Ich stellte fest, dass der Computer, der vor zwei Tagen so sorgfältig aufgebaut worden war, nirgendwo zu finden war. Es war der helle Wahnsinn. Jemand erwähnte, dass die Leute über vier Stunden im Voraus begonnen hatten, sich für den Vortrag anzustellen, und dass

der Wartebereich zum Brechen voll war. In der Zwischenzeit rannten wir wie kopflose Hühner herum und durchkämmten die Räume hinter der Bühne auf der Suche nach dem Rechner.

Es war ein normales Desktop-System, auf dem Star Office, eine der Linux-Officelösungen, installiert war. Und eigentlich hätte ich nur meine Diskette einzulegen und anzufangen brauchen. Alles war so aufgebaut, dass man noch nicht einmal Kabel anschließen musste. Aber der Computer war verschwunden. Offensichtlich war die Maschine aus irgendeinem Grund falsch beschriftet und zurückgeschickt worden. Glücklicherweise hatte ich meinen Laptop dabei, auf dem die aktuellen Folien für den Vortrag gespeichert waren und Star Office installiert war.

Allerdings befanden sich auf meinem Laptop nicht alle benötigten Fonts. Das bedeutete, dass auf allen Folien die letzte Zeile fehlte. Als mir das klar wurde, dachte ich: *Was soll's? Ich werde das lebend überstehen.* Dann mussten wir all die Kabel anschließen. Noch bevor alles aufgebaut war, fingen sie an, die Leute hereinzulassen. Ich meine, das muss man sich mal vorstellen. Ich war da oben und versuchte immer noch, die Sache zum Laufen zu bringen, als ein Menschenschwall sich in das Auditorium ergoss, jeden verfügbaren Sitzplatz belegte und dann die Stehplätze an den Seiten füllte. Glücklicherweise bekam ich die stehende Ovation, *bevor* ich den Mund aufmachte.

Ich begann mit einer lahmen Anspielung auf den Anwaltswitz, mit dem Bill Gates seinen Vortrag eröffnet hatte. Ich ging in einem einzigen Satz auf das Entwicklungsziel der damals geheimnisumwitterten Transmeta Corporation ein. In der Presse war lang und breit darüber spekuliert worden, dass ich die Rede bei der Comdex als Gelegenheit nutzen würde, (endlich) den Transmeta-Chip anzukündigen. Aber so weit waren wir einfach noch nicht. Der Hauptteil meines Vortrags erschöpfte sich deshalb darin, dass ich die Vorteile von Open Source aufzählte. Ich war nicht in der Stimmung, so viele Witze zu machen wie sonst. Irgendwann fing Daniela - die mit Tove und Patricia in der ersten Reihe saß - so laut zu schreien an, dass man es wahrscheinlich in allen Kasinos und Nachtclubs von Las Vegas hören konnte.

Dieser Vortrag wird nicht in die Geschichte der großen Reden eingehen. Später versuchte jemand mich mit der Bemerkung aufzumuntern, auch Bill Gates sei bei seinem gestrigen Vortrag auf der gleichen Bühne

sichtlich nervös gewesen. Seine Bühnentechnik hatte reibungslos funktioniert. Er hatte ein anderes Problem: Das amerikanische Justizministerium saß ihm im Nacken. Ich schätze, verglichen damit war ich noch glimpflich davongekommen.

Die Strategie hätte der Vorlesung „Journalismus für Anfänger“ entnommen sein können: Such dir die Person, die schon am längsten ansteht, um Linus' Keynote-Rede zu hören und leiste ihm (denn ohne jeden Zweifel ist diese Person männlichen Geschlechts) beim Warten Gesellschaft. Welche bessere Möglichkeit gäbe es, sich einen Eindruck von den wildgewordenen Massen zu verschaffen, die Linus verfolgen wie eine in Werbe-T-Shirts gewandete Gottheit?

Um 17 Uhr fahre ich mit einem Lift in das Woodstock der Geeks hinunter. An der Spitze einer riesigen Warteschlange steht ein fiebriger Informatikstudent vom Walla Walla College, der bereitwillig zulässt, dass ich mich ihm anschließe. Er steht schon seit zweieinhalb Stunden an, um Linus zu sehen, und wird noch weitere zweieinhalb Stunden warten, bis man ihn endlich ins Auditorium lässt. Seine Kommilitonen, die hinter ihm in der Schlange warten, kamen etwa eine halbe Stunde nach ihm an. Sie sind mit einem ihrer Professoren aus dem Bundesstaat Washington angereist und schlafen in der Sporthalle einer örtlichen High-School. Meinem Eindruck zufolge hat jeder von ihnen bereits seine eigene Web-Design-Firma gegründet. Sinnvollerweise haben sie das Universum der Erwachsenen in zwei Gruppen aufgeteilt - Hacker und Anzugträger - und weisen einander ununterbrochen auf Angehörige der zweiten Spezies in der anschwellenden Schlange hin. „Mann, schau dir bloß die ganzen Anzugträger an“, sagen sie, ungefähr so, wie ihre Delta-Tau-Chi-Pendants in den Frühjahrsferien einen Strand mustern und bemerken: „Mann, schau dir bloß die ganzen Weiber an.“ Ansonsten führen sie sich kaum anders auf als ihre Delta-Tau-Chi-Pendants: Sie schlagen die erhobenen Handflächen gegeneinander und tauschen Unfreundlichkeiten aus, die allerdings alle etwas mit Motherboards und Gigabytes zu tun haben.

Und dann reden sie über Linus. Sie sprechen seinen Namen aus, als würde er in Großbuchstaben geschrieben: „LINUS würde nie bei einer Firma arbeiten, die ihre Software nicht frei zugänglich macht. Das würde er einfach nicht tun.“ Sklavisch verfolgen sie Websites wie

Slashdot, auf denen Gerüchte über das geheimnisumwitterte Tun und Treiben bei Transmeta gehandelt werden wie anderswo die obskuren Details aus dem Liebesleben eines Hollywood-Starlets. Diese Manie, diese Mischung aus Spekulation und Faszination kennzeichnet nicht nur die leidenschaftlichen Groupies, die als Erste angekommen waren.

Ich gehe zur Toilette, stelle mich an das einzige freie Urfinal und werde Zeuge des folgenden Dialogs.

„Diese Rede wird, verglichen mit der Keynote von Gates, total langweilig sein“, sagt der Typ zu meiner Rechten.

„Jetzt mach aber mal einen Punkt“, antwortet der andere Typ. „Was erwartest du denn? Linus ist schließlich ein Hacker, kein Anzugträger.“

Als wir endlich im Auditorium angelangt sind, finden wir uns seltsamerweise nicht ganz vorn, sondern im mittleren Teil ziemlich weit hinten wieder. Einen Moment lang vergisst mein Mit-Schlangesteher aus Walla Walla seine Vorfreude, seinen Helden live zu sehen, und bekommt einen Wutanfall, weil er nicht in der ersten Reihe sitzt, wie er es eigentlich verdient hätte. Aber es dauert nicht lange, und er weist mich auf die Anzugträger im Publikum hin. Obwohl wir an die 70 Meter vom Podium entfernt sitzen, ist es möglich, einen Blick auf Linus zu erhaschen, der auf der verdunkelten Bühne an einem Computer sitzt. Ein paar Offizielle stehen um ihn herum, während er wie wild auf seiner Tastatur tippt. Was mag das bedeuten? Vielleicht eine Softwarevorführung in letzter Minute?

Schließlich verlassen Linus und die anderen die Bühne. Irgendjemand stellt den Executive Director von Linus International Maddog (Jon Hall) vor. Die Aufregung meines Begleiters aus Walla Walla steigt sichtlich. „Schau dir bloß den Bart an“, sagt er. Dann verkündet Maddog, wie sehr er sich darüber freut, einen Mann vorstellen zu können, der wie ein Sohn für ihn ist. Linus kommt wieder auf die Bühne, wo Maddog ihn mit einer herzlichen, haarigen Umarmung empfängt.

Selbst ich auf meinem Billigplatz ziemlich weit hinten kann seine Nervosität erkennen.

„Ich wollte mit einem Anwaltswitz beginnen, aber der wurde schon erzählt“, sagt er in Anspielung auf die gelungene Rede, die der durch den Antitrustprozess geplagte Bill Gates am Vorabend gehalten hat. „Kennt jemand ein paar gute Anwaltswitze?“

Als Nächstes geht er in einem einzigen Satz auf Transmetas geheime Mission ein. Der Rest seiner Rede besteht daraus, die Sätze abzuspuhlen, die hoch über seinem Kopf als Folien gezeigt werden, Aussagen über die wachsende Bedeutung von Opera Source. Nichts Überraschendes. Nichts Neues.

Das Ganze wird in erschöpfter-aber-gutgelaunter Gleichförmigkeit vorgetragen. Einmal schreit eine seiner Töchter.

Mitten im Satz sagt er: „Das ist mein Kind.“ Wenn man auf den Monitor schaut, sieht man, wie sich die Bühnenbeleuchtung in den Schweißperlen auf seiner Stirn widerspiegelt.

Anschließend stehen die Zuhörer Schlange, um ihm Fragen zu stellen. Ohne zu zögern, lehnt er es ab zu sagen, welche Linux-Textverarbeitung er bevorzugt.

Als ihn jemand fragt, wie viele ausgestopfte Pinguine er zu Hause hat, antwortet er: „Eine ganze Menge.“ Ein Zuhörer fragt ihn, wie ihm das Leben in Kalifornien gefällt, und er antwortet mit einer Rhapsodie über das Wetter.

„Es ist November, und ich trage immer noch Shorts. In Helsinki hätte mich das schon meine Kronjuwelen gekostet.“ Ein Fan geht an das Mikrophon, das für Fragen aus dem Publikum aufgestellt wurde, und verkündet einfach:

„Linus, du bist mein Held.“ Worauf er reagiert, als hätte er diese Feststellung schon eine Million Mal gehört und ebenso oft beantwortet: „Danke.“

Am Ende der Fragestunde strömen die Leute zu Hunderten in den Bereich unterhalb des Podiums, wo Linus inzwischen steht und so viele Hände schüttelt, wie er nur kann.

IX.

Ist die Linuxrevolution vorbei?

Von Scott Berinato, PC Week

„Vielen Dank für Ihren Anruf. Die Revolution ist vorbei. Wenn Sie weitere Informationen über Linux erhalten möchten, drücken Sie bitte die Eins..."

Es sieht so aus, als hätte Linus Torvalds einen Agenten, und das muss heißen, dass diese ganze Linux-Sache zum Mainstream geworden ist, so dass du die Revolution vergessen und dich wieder deiner Arbeit auf deinem Windows-Desktop zuwenden kannst.

Es gab einmal eine Zeit, da konntest du als Journalist den Erfinder des Betriebssystems Linux in seinem Büro in der als Mantel-und-Degen-Drama vermarkteten Transmeta Corporation direkt unter seiner Durchwahl anrufen und mit dem wohl vertrauten entschiedenen „Torvalds" des Helden selbst am anderen Ende rechnen. Er war geduldig und beantwortete deine Fragen. Er sagte dir, wenn er keine Zeit hatte. Manchmal sagte er dir, deine Fragen zur Programmierung seien anfängerhaft und wenig hilfreich. Aber er ging ans Telefon.

Wenn du heute bei Transmeta anrufst und seine Durchwahl eintippst, begrüßt dich eine freundliche Frauenstimme. „Danke, dass Sie Linus Torvalds angerufen haben. Dieser elektronische Anrufbeantworter nimmt keine Nachrichten entgegen. Um Linus Torvalds zu kontaktieren, senden Sie bitte ein Fax an ..."

Was? Und dann begreifst du. Er wird dich nicht zurückrufen. Ihm reicht's. Er ist eine Berühmtheit geworden, und ein schnelles Interview mit ihm zu vereinbaren ist heute genauso, als wollte man ein schnelles Interview mit jener anderen großen Berühmtheit der Computerbranche vereinbaren. Die Frauenstimme rattert eine Faxnummer herunter, und dir fällt ein, du könntest die alte 0-#-Kombination wählen und die Rezeption ...

„Unsere Rezeptionisten nehmen keine Nachrichten für ihn an und sind nicht für seinen Terminkalender zuständig." La-di-da. Sie ist überaus liebenswürdig. Die schlimmste Art. „Aber sie werden ihm gerne Ihr Fax übermitteln." Hm. Und Bill wird gerne Microsoft zerschlagen, um David Boies zu besänftigen.

Okay, die Linux-Revolution ist nicht vorbei, aber wie bei jeder Revolution werden die echten Fanatiker zunehmend durch Mainstream-Sympathisanten abgelöst. Der urbane Punk Rock wurde durch die New-Wave-Musik der Vorstädte ersetzt. Aus den von den Engländern ausgebeuteten Steuerzahlern der amerikanischen Kolonien wurden wohlhabende Landbesitzer. (Übrigens versuchten die wohlhabenden Landbesitzer später ihrerseits, den Pionieren des Westens eine Whiskey-Steuer aufs Auge zu drücken, die sich kaum von der Teesteuer unterschied, die ihnen selbst ein paar Jahre früher abverlangt worden war.)

Wahrscheinlich war es wirklich höchste Zeit, dass Linus anfang, sich rarer zu machen. Angesichts der zahlreichen Presseanrufen und der verrückten Vielfalt der Themen, zu denen er Stellung nehmen sollte, war das unvermeidlich.

Schauen Sie sich sein Presseinterview bei der Linux World Expo in San Jose Anfang des Monats an. Es blieb Torvalds, der sich auf das Presseinterview eingelassen hatte, weil er einfach nicht die Zeit fand, die unzähligen Einzelanfragen zu beantworten, nicht erspart, zuerst eine Reihe mittlerweile vertrauter Antworten auf vertraute Fragen abzuspielen. Kann Ctpen Source in der Geschäftswelt funktionieren? Streben Sie an, den Softwarebereich auf ähnliche Weise wie Bill Gates zu beherrschen? Was halten Sie von Microsoft? Was ist Open Source? Was ist Linux? Warum ein Pinguin?

Torvalds' Antworten begannen an diesem Punkt den Ton-Konserven zu ähneln, die Spitzensportler von sich zu geben pflegen. Denken Sie an Tim Robbins in dem film *Annie's Männer*. „Ich werde dort hinausgehen und 110 Prozent geben, um dem Team zu helfen ...“

Viele Journalistenfragen waren aber nicht nur redundant, sie betrafen auch ein weites Feld jenseits der technischen Welt. So wurde das finnische Wunder gefragt, wie es gedachte, den Markt der kleinen und mittelgroßen Unternehmen zu erobern. (Typische Torvalds-Antwort: „Ich persönlich habe nie versucht, jemanden zu erobern.“) Zwei Fragen später wollte ein eifriger, Ich-rolle-diesen-Open-Source-Krampf-ganz-anders-auf-Reporter von Torvalds wissen, was er von Unternehmen halte, die für landwirtschaftliche Zwecke Genome patentieren lassen. (Typisches Torvalds-Soundbite: „Ich bin gespalten, was Patente anbetrifft. Es gibt gute schlechte und wirklich schlechte.“)

Programmierer, lasst euch sagen: Wenn jemand anfängt, euch nach Genomen für die Landwirtschaft zu befragen, ist es vermutlich an der Zeit, einen Agenten anzuheuern.

Deshalb ist es vielleicht doch ganz gut, dass Linus nicht mehr ans Telefon geht. Trotzdem werden wir Torvalds' Offenheit und Selbstironie vermissen, die auf Journalisten, die regelmäßig einen trockenen Hals von dem Flugzeugluft-Marketing bekommen, mit dem die meisten Unternehmen sie anblasen, so erfrischend natürlich wirkt. Und falls Faxe seinen Schreibtisch wirklich erreichen und er tatsächlich auf unsere Fragen antworten sollte, behält er hoffentlich den torvaldsschen Ton bei. Denn wenn die PR-Stimmen mit ihrer falschen Liebenswürdigkeit übernehmen, wird der Linux-Hype nicht mehr annähernd so viel Spaß machen. '

Okay, ich denke, ich schulde Mr. Berinato eine Erklärung, wenn auch keine Entschuldigung.

Jeder, der diesen Beitrag liest, würde annehmen, ich sei infolge der zunehmenden Belastungen meiner Rolle als Chef-Nerd zum Arschloch mutiert. Aber das stimmt nicht. Ich war schon immer ein Arschloch.

Lassen Sie mich am Anfang beginnen. Ich finde, Anrufbeantworter sind ein Teufelszeug. Sie sind das perfekte Beispiel einer schlechten Technik. Tatsächlich sind sie die schlechteste Technik überhaupt, und ich hasse sie abgrundtief. Bei Transmeta gab es anfangs ein Anrufbeantwortersystem, das für jeden Mitarbeiter bis zu 20 Minuten Nachrichten speicherte. Danach bekamen Anrufer zu hören, die **Mailbox sei voll, und sie sollten bitte die Rezeption anrufen**. Meine Mailbox war immer voll.

Ich denke, es waren die Journalisten, die die Gegenreaktion auslösten. **Sie nervten die Rezeptionisten bei Transmeta, weil meine Mailbox voll war. Nach den ersten paar hundert Mal hatten die Rezeptionisten dann allmählich die Nase voll**. Sie wussten, ich hatte an diesen Anrufen **kein Interesse, und sie hatten keine Lust, den Leuten** sagen zu müssen, sie sollten sich zum Teufel scheren.

Also ging ich dazu über, Nachrichten zu löschen, ohne sie **abzuhören, einfach damit die Leute am Empfang sich nicht ärgerten. Die meiste Zeit hörte ich mir meine Nachrichten ohnehin nicht** an. Erstens murmelten die Leute erfahrungsgemäß ihre Telefonnummer in den Ansa-getext, so dass ich ihre Nachricht fünfzehnmal anhören musste, nur um das Gesagte zu verstehen. Und außerdem weigere ich mich, Leute zurückzurufen, wenn ich keinen Grund sehe, sie zurückzurufen. Die Leute hatten ein warmes wohliges Gefühl, weil sie eine Nachricht hinterlassen hatten. Bis sie merkten, dass ich sie nicht zurückrufen würde.

An diesem Punkt riefen sie bei der Rezeption an. Die Rezeptionisten wussten nicht, wie sie damit umgehen sollten, und deshalb sagte ich **ihnen, sie sollten dem Anrufer sagen, er könnte mir ein Fax schicken**. Faxe lassen sich genauso leicht ignorieren wie Nachrichten auf **dem Anrufbeantworter, aber mit einem Fax kannst du, wenn du willst, zumindest die Nummer feststellen. Nicht, dass ich das je gewollt hätte**. Zuerst legten die Rezeptionisten Anrufern höflich nahe, sie mögen mir **bitte ein Fax schicken**. Irgendwann wurde den Leuten dann klar, dass ich das Fax nicht gelesen hatte, und sie riefen eine Woche später zurück und beklagten sich, sie hätten mir bereits ein Fax geschickt. **Und damit** saßen die Rezeptionisten schon wieder zwischen zwei Stühlen. Es war nicht ihre Aufgabe, meine Anrufe entgegenzunehmen.

Vergessen Sie Mr. Berinatos großherzige Beschreibung meiner Person in den guten alten Tagen vor Linux' Höhenflug: In Wirklichkeit war ich schon immer ein Arschloch. Das ist nichts Neues.

Die Faxlösung war nicht von langer Dauer. Am Ende wurde meine Mailbox so eingerichtet, dass man keine Nachrichten darauf hinterlassen konnte. Inzwischen hatte Transmeta eine PR-Person eingestellt, die sich freiwillig anbot, Anfragen an mich zu beantworten. Diese Leute haben das gelernt, sagt man mir. Trotzdem verlangt man nach wie vor von mir, Journalisten immer zurückzurufen; auch wenn ich nicht mit ihnen sprechen möchte, vermittele es den Leuten ein warmes wohliges Gefühl, wenn ich mich wenigstens bei ihnen melde. Dazu nur so viel: Ihre warmen wohligen Gefühle lassen mich kalt.

Okay. Ich gehe selbst ans Telefon, wenn jemand zufällig dann anruft, wenn ich an meinem Schreibtisch sitze. Aber das ist weder als Versuch, zugänglich wirken zu wollen, noch als politisches Statement zu werten. Open Source hat nie bedeutet, dass ich zugänglicher bin als andere Leute. Es hat nie bedeutet, dass ich moralischer bin als andere Leute. Es hat nie bedeutet, dass ich offener für die Vorschläge anderer bin. Das war nie der Punkt. Der Punkt ist, dass die Leute mich links liegen lassen können, wenn ich mich als der schwärzeste Teufel der Hölle oder als grundböse erweisen sollte: weil sie ihr Ding genauso gut alleine machen können. Es geht nicht darum, dass ich offen bin, es geht darum, dass sie die Macht haben, mich links liegen zu lassen. Das ist es, was wirklich zählt.

Es gibt keine „offizielle“ Linux-Version. Es gibt meine Version, und es gibt die Versionen aller anderen. Tatsache ist, die meisten Leute vertrauen meiner Version und betrachten sie als die de facto offizielle Version, weil sie gesehen haben, dass ich seit neun Jahren daran arbeite. Ich war der Erste, der Urvater sozusagen, und im Großen und Ganzen herrscht Einigkeit darüber, dass ich einen guten Job mache. Aber nehmen Sie an, ich rasiere meinen Kopf so, dass 666 darauf zu lesen ist, und sage: „Kniet vor mir nieder, anderenfalls mache ich euch alle.“ Sie würden mir nur ins Gesicht lachen und sagen: „Okay, wir nehmen einfach diesen kleinen Kernel und tun damit, was wir für richtig halten.“

Die Leute vertrauen mir. Aber sie tun das nur aus einem einzigen Grund: Bisher habe ich mich als vertrauenswürdig erwiesen.

Das heißt nicht, dass ich bereit bin, den Anrufbeantworter abzuhören - oder jemandem zuzuhören, der mich zufällig am Telefon erwischt. Ich habe nie darauf Wert gelegt, dass die Leute in mir den guten Menschen sehen, der gerne jeden Anruf und jede E-Mail beantwortet. Und wenn wir schon beim Thema sind: Ich finde diese Geschichten, die mich als aufopfernden Mönch oder Heiligen hinstellen, für den Geld einfach keine Bedeutung hat, höchst seltsam. Ich versuche seit Jahren, diesem Mythos entgegenzuwirken, aber in der Presse ist davon nichts zu merken. Ich will nicht der Mensch sein, zu dem mich die Presse machen will.

Tatsache ist, ich habe das Image des aufopfernden Mönchs immer verabscheut, weil es so uncool ist. Es ist ein langweiliges Image. Und es stimmt nicht.

X.

Als ich aus meinem Zimmer herauskroch und mich unversehens im Rampenlicht wiederfand, musste ich schnell die Wahrheiten des Lebens lernen, die andere Leute wahrscheinlich auf dem Weg zum Kindergarten mitbekommen. Ich hätte zum Beispiel nie damit gerechnet, wie lächerlich ernst die Leute mich oder jeden meiner Schritte nehmen würden. Die beiden folgenden Situationen sind typische Illustrationen dieses Themas mit Variationen.

Damals an der Uni hatte ich eine Root-Benutzerkennung auf meinem Rechner. Mit jeder Benutzerkennung ist ein Name verbunden. Der Name dient lediglich Informationszwecken. Deshalb nannte ich auf meinem Rechner die Kennung des Root-Benutzers Linus God Torvalds. Ich war der Gott dieser Kiste, die in meinem Büro in der Uni stand. Rechtfertigt das wirklich einen Aufstand?

Also: Wenn jemand einen Rechner, der unter Linux oder Unix läuft, „fingert“, will er sehen, wer in diesen Rechner eingeloggt ist. Seit dem Aufkommen von Firewalls gehört dieses Fingern weitgehend der Vergangenheit an. Aber vor Jahren fingerten die Leute andere Rechner, um zu sehen, ob der Anwender eingeloggt war oder seine E-Mails gelesen hatte. Das Fingern bot darüber hinaus eine Möglichkeit, sich den „Plan“ einer Person anzusehen, persönliche Informationen, die der Betreffende auf seinem Rechner gepostet hatte, so eine Art Vorläufer der Webseite. In meinem Plan war immer die jüngste Version des Linux-Kerns zu finden. Das Fingern meines Rechners stellte daher eine Möglichkeit dar, die gerade aktuelle Version kennen zu lernen. Manche Leute hatten den Vorgang sogar automatisiert. Sie fingerten mich einmal in der Stunde, um sich über Versionsänderungen auf dem Laufenden zu halten. Wann immer mich jemand fingerte, sah er, dass ich die Root-Benutzerkennung Linus God Torvalds hatte. Anfangs stellte

das kein Problem dar. Dann bekam ich auf einmal Mails von Leuten, die den Namen als Blasphemie empfanden. Deshalb änderte ich ihn. Leute dieser Art nehmen sich selbst zu ernst. Und das geht mir auf den Geist.

Dann gab es natürlich noch den Vorfall in North Carolina. Mann! Das war wirklich schlimm. Ein jüngst erschienen Buch über Red Hat spielt das Ganze zu einem internationalen Zwischenfall mit potenziell katastrophalen Ausmaßen hoch. Dabei war eigentlich nicht viel passiert.

Ich war eingeladen worden, einen Vortrag bei einem Linux-Anwendertreffen zu halten, das Red Hat in seiner Firmenzentrale in Durham veranstaltete. Der Saal war brechend voll. Als ich auf die Bühne kam, erhoben sich alle von ihren Plätzen und jubelten mir zu. Die ersten Worte aus meinem Mund waren das Erstbeste, was mir in den Sinn kam:

„Ich bin euer Gott.“

Verdammt noch mal, das sollte ein Witz sein!

Ich wollte damit nicht sagen: „Ich bin zutiefst davon überzeugt, dass ich euer Gott bin und daran solltet ihr immer denken.“ Ich wollte

damit sagen: „Okay, okay, okay. Ich weiß, ich bin euer Gott. Jetzt setzt euch bitte einfach hin und wartet mit dem Jubel, bis ihr gehört habt, was ich zu sagen habe, obwohl ich euren Vorschussjubiläum aufrichtig zu schätzen weiß.“

Ich kann nicht glauben, dass ich dieses Ereignis freiwillig wieder aufleben lasse.

Nach diesen vier Begrüßungsworten war es einen Moment lang ganz still. Ein paar Stunden später waren meine einleitenden Worte das Thema von Newsgroup-Postings geworden. Ich gebe es zu: Sie waren eine Entgleisung. Aber keine Absicht. Wahrscheinlich war es einfach

meine Art, mit der Peinlichkeit umzugehen, dass die Leute sich applaudierend von ihren Plätzen erheben, nur weil du zum Rednerpult gehst.

Die Leute nehmen mich zu ernst. Sie nehmen vieles zu ernst. Aber das ist - diese Lektion habe ich in meinen Jahren als Galionsfigur von Linux gelernt - noch nicht das Schlimmste: Manchen Leuten reicht es nicht, wenn sie selbst die Dinge zu ernst nehmen. Sie sind erst glücklich, wenn sie andere davon überzeugen können, der gleichen Obses-

sion zu erliegen. Diese Beobachtung hat sich zu einer meiner wiederkehrenden Lieblingsklagen über das Leben entwickelt.

Haben Sie sich je die Zeit genommen, darüber nachzudenken, warum Hunde so an ihren Menschen hängen? Nein, nicht weil ihre Besitzer sie alle sechs Wochen zum Hundefriseur bringen und gelegentlich ihre Hinterlassenschaften auf dem Gehweg beseitigen. Sondern weil Hunde sich gerne sagen lassen, was sie tun sollen. Das gibt ihrem Leben einen Sinn. (Das gilt umso mehr, als viele von ihnen arbeitslos sind - kastriert und damit ihrer Aufgabe als Produzenten neuer Hundegenerationen enthoben. Außerdem sind ihre festverdrahteten Fähigkeiten wie das Aufspüren von Nagetieren, von wenigen Ausnahmen abgesehen, nicht sonderlich gefragt.) Als Mensch bist du der Anführer des Rudels und sagst dem Hund, wie er sich benehmen soll. Deinen Befehlen zu folgen, ist seine Leidenschaft. Er tut das gern.

Leider sind Menschen genau so gestrickt. Die Leute wollen, dass jemand ihnen sagt, was sie tun sollen. Das ist in unserem Kernel so angelegt. Jedes soziale Wesen muss so sein.

Das heißt nicht, dass du untertänig bist. Es heißt nur, dass du dazu neigst, mitzumachen, was andere dir vorleben.

Daneben gibt es Leute mit eigenständigen Meinungen, Leute mit so starken Überzeugungen, dass sie sagen: „Nein, das mache ich nicht mit.“ Diese Leute werden zu Anführern. Es ist einfach, ein Anführer zu werden. (Es muss einfach sein. Schließlich habe ich es auch geschafft, oder?) Leute, die in bestimmten Bereichen keine Überzeugungen haben, überlassen es nur allzu bereitwillig den Anführern, Entscheidungen für sie zu treffen und ihnen zu sagen, was sie tun sollen.

Keine Frage, jeder Mensch hat das Recht, zu tun, was ein anderer ihm sagt, den er sich als Oberhaupt erwählt hat. Dagegen gibt es nichts einzuwenden. Ich persönlich finde das zwar deprimierend, aber sei's drum. Wirklich unakzeptabel finde ich es dagegen, wenn Leute -Anführer oder Mitläufer - versuchen, anderen ihre eigene Weltsicht aufzuzwingen. Das ist nicht nur deprimierend, sondern erschreckend. Es ist deprimierend, dass Leute bereit sind, allem und jedem zu folgen, unter anderem auch mir. Und es ist erschreckend, dass Leute diese ihre Folgensbereitschaft - wenn es dieses Wort überhaupt gibt - anderen aufzwingen wollen, unter anderem auch mir.

Vergessen Sie die gepflegten RoboterBekehrer, die immer dann an Ihre Tür klopfen, wenn Sie am Computer sitzen und sich auf ein schwieri-

ges technisches Problem konzentrieren oder wenn die Kinder endlich eingeschlafen sind und sich erotische Gefühle in Ihnen zu regen beginnen. In der Open-Source-Szene gibt es noch schlimmere Leute: Jene Eiferer, die glauben, jede Innovation solle unter der GPL lizenziert werden („GPLt“ werden, wie es im Hackerjargon heißt). Richard Stallman will jede Software frei zugänglich machen. Für ihn ist das ein Politikum, und er will die GPL als Möglichkeit nutzen, den OpenSource-Gedanken voranzutreiben. Er sieht dazu keine Alternative. Aber ich habe Linux, um ehrlich zu sein, nicht aus so hehren Gründen als Open-Source-Software freigegeben. Mir ging es um Rückmeldungen. Und außerdem war das das übliche Verfahren. In den frühen Tagen des Computers wurde ein Großteil der Arbeit an Hochschulen oder militärischen Forschungseinrichtungen erledigt und letztlich sehr offen gehandhabt. Es war ganz normal, seinen Quellcode einer anderen Hochschule zu überlassen, wenn man darum gebeten wurde. Richard war der Erste, der sich, nachdem er von Projekten, die er liebte, abgeschnitten worden war, bewusst für Open Source einsetzte.

Ja, es bringt enorme Vorteile, seine Technik zu öffnen und nach den gleichen Bedingungen zugänglich zu machen wie Linux und zahlreiche andere Innovationen. Um eine Ahnung von diesen Vorteilen zu bekommen, müssen Sie sich nur die relativ niedrigen Qualitätsstandards jedes geschlossenen Softwareprojekts ansehen. Die GPL und das Open-Source-Modell unterstützen die Entwicklung der bestmöglichen Technik. So einfach ist das. Sie verhindern auch, dass Technik gehortet wird. Und sie stellen sicher, dass niemand, der an einem Projekt oder einer Technik interessiert ist, von Weiterentwicklungen ausgeschlossen wird.

Das ist kein unwesentlicher Punkt. Stallman, der es verdienen würde, dass man ihm zu Ehren ein Denkmal für die Schaffung der GPL errichtet, stieß das Phänomen der freien Software hauptsächlich an, weil er den Zugang zu einer Reihe interessanter Entwicklungsprojekte verlor, als sie sich von der offenen akademischen Welt des MIT in die auf ihr Eigentum bedachte Welt der Unternehmen verlagerten. Das bekannteste dieser Projekte war die LISP-Maschine. LISP entstammte ursprünglich der KI-Szene. Dann hielt irgendjemand es für so gut, dass er ein Startup-Unternehmen gründete, um es zum kommerziellen Erfolg zu führen und Geld damit zu verdienen. Das passiert an den Hochschulen ständig. Aber Richard gehörte nicht der kommerziellen

Szene an. Als LISP 1981 das kommerzielle Projekt einer Firma namens Symbolics wurde, war er plötzlich außen vor. Und um seiner Frustration die Krone aufzusetzen, warb Symbolics viele seiner Kumpels aus dem AI-Lab ab.

Das Gleiche passierte ihm noch öfter. So wie ich es sehe, entspringt seine Motivation, den Open-Source-Gedanken zu promoten, weniger einer anti-kommerziellen Haltung als vielmehr einer anti-exklusiven. Für ihn bedeutet Open Source, nicht ausgeschlossen zu werden. Jeder soll weiter an einem Projekt arbeiten können, unabhängig davon, wer es kommerziell verwertet.

Die GPL ist insofern eine wunderbare Sache, als sie dafür sorgt, dass jeder mitspielen kann. Sie müssen sich den enormen Fortschritt nur mal vorstellen, den das für die Menschheit bedeutet! Aber heißt das, dass jede Innovation GPLt werden muss?

Keineswegs! Das ist die Abtreibungsfrage der Technologie. Der einzelne Neuerer sollte selbst entscheiden können, ob er sein Projekt GPLen will oder lieber ein konventionelleres Urheberrecht verwenden möchte. Was mich an Richard so verrückt macht, ist seine SchwarzWeiß-Malerei. Sie führt zu unnötigen politischen Spaltungen. Er kann nie den Standpunkt eines anderen verstehen. Wenn es um Religion ginge, würde ich ihn als religiösen Fanatiker bezeichnen.

Tatsächlich kommen auf der Liste der Ärgernisse gleich nach religiösen Eiferern, die an meine Tür klopfen und mir vorschreiben, was ich glauben soll, Leute, die an meine Tür klopfen (oder mich mit E-Mails bombardieren) und mir vorschreiben, wie ich meine Software lizenzieren soll. Dieses Thema sollte kein Politikum sein. Die Leute sollten ihre eigenen Entscheidungen treffen können. Es ist eine Sache, jemanden ans Herz zu legen, seine Software zu GPLen, und es dabei zu belassen. Es ist etwas ganz anderes, das einzufordern. Ich finde es einfach verfehlt, wenn Leute sich darüber beklagen, dass ich für ein kommerzielles Unternehmen arbeite, das seine Arbeit nicht komplett nach der GPL lizenziert. Das geht sie einfach nichts an.

Was mich an Richard so unglaublich aufregt, ist nicht seine Überzeugung, Linux müsste, weil sein Kernel sich anfangs auf Anwendungen aus dem GNU-Softwareprojekt stützte, von Rechts wegen „gnu/ Linux“ heißen. Es ist auch nicht sein offen zur Schau getragener Groll, dass ich eine Galionsfigur für Open Source bin, obwohl er schon Code zugänglich machte, als ich noch in einem Wäschekorb schlief. Nein,

ich finde ihn so unerträglich, weil er sich ständig über Leute beklagt, die nicht die GPL verwenden.

Aus der Ferne bewundere ich Richard aus einer Reihe von Gründen. Und vermutlich nötigen mir Menschen wie Richard, die starke moralische Meinungen vertreten, sogar Respekt ab. Aber warum können sie ihre Ansichten nicht für sich behalten? Ich verabscheue nichts so sehr wie Leute, die mir sagen, was ich tun soll und was nicht. Mir sind Menschen, die sich ein Mitspracherecht bei meinen persönlichen Entscheidungen anmaßen, zutiefst zuwider. (Außer, vielleicht, meine Frau.)

Im Lauf der Linux-Entwicklung haben Experten wie Eric Raymond die Vermutung aufgestellt, der Erfolg des Betriebssystems und die Langlebigkeit der Open-Source-Entwicklung sei teilweise meinem Pragmatismus und meiner Fähigkeit zu verdanken, bei Streitigkeiten keine Partei zu ergreifen. Während Eric der wohl beste Verfechter des Open-Source-Phänomens ist (und während ich seine Befürwortung von Waffen total und auf der ganzen Linie ablehne), glaube ich, dass er mit dieser Interpretation etwas daneben liegt. Es ist nicht so, dass ich keine Partei ergreifen würde. Ich lehne ganz einfach Leute ab, die ihre Moralvorstellungen anderen aufzuzwingen versuchen. Anstelle des Wortes „Moral“ könnte hier ebenso gut stehen „Religion“, „Rechnervorlieben“, was immer Sie wollen.

Wenn es falsch ist, anderen seine Moralvorstellungen aufzuzwingen, ist der nächste Schritt - Moralvorstellungen zu *institutionalisieren* -doppelt falsch. Ich glaube an das Wahlrecht des Einzelnen, und finde deshalb, ich sollte, was moralische Fragen anbetrifft, meine Entscheidungen selbst treffen.

Ich will selbst für mich entscheiden können. Ich bin total gegen unnötige Regeln, die uns von der Gesellschaft auferlegt werden. Ich glaube daran, dass jeder in der Privatheit seines eigenen Zuhauses tun und lassen können sollte, was er möchte, solange er damit niemandem schadet. Jedes Gesetz, das etwas anderes vorschreibt, ist krank. Aber es gibt solche Gesetze. Manche davon finde ich erschreckend, vor allem einige von denen, die für Kinder und Schulen gelten. Stellen Sie sich vor, jemand kommt auf die Idee, Regeln über die Vermittlung der Evolutionstheorie aufzustellen, und liegt mit seinen Ansichten falsch. So etwas finde ich erschreckend. Hier erhebt das soziale Gewissen sein hässliches Haupt in Bereichen, in denen es wirklich nichts zu suchen hat.

Gleichzeitig glaube ich persönlich, dass es noch etwas Wichtigeres gibt als mich und meine individuellen moralischen Entscheidungen: nicht die Menschheit, sondern die *Evolution*. In dieser Hinsicht sollten meine individuellen Entscheidungen tatsächlich gesellschaftliche Belange in Betracht ziehen. Aber das ist wahrscheinlich auch so in uns angelegt. Ich denke, die Biologie - die Evolution - hat dafür Sorge getragen, dass Menschen gesellschaftliche Erwägungen mit berücksichtigen. Anderenfalls würde es uns längst nicht mehr geben.

Die einzige andere Sache, die Wutausbrüche rechtfertigt, sind Leute mit allzu ausgeprägtem missionarischen Eifer. Es gibt einfach keinen Grund, andere bekehren zu wollen und sich dabei so selbstgerecht zu gebärden.

Und ich klinge wie einer von ihnen.

Aber in diese Falle gerät man leicht, wenn die Leute einen plötzlich viel zu ernst nehmen.

XI.

Die Amerikaner machen einen großen Aufstand um den 17. März (St. Patricks Day), den 5. Mai (Cinco de Mayo) und den 12. Oktober (Columbus Day). Dagegen verläuft der 6. Dezember, der finnische Unabhängigkeitstag, wie jeder Finne weiß, weitgehend ereignislos.

Die meisten Leute in Finnland begehen den finnischen Unabhängigkeitstag auf die gleiche Weise wie alles andere: Sie feiern bis zum Umfallen. Sie feiern am Vorabend bis zum Umfallen - selbst für finnische Verhältnisse - und hängen dann fast den ganzen Nationalfeiertag abgeschlafft vor dem Fernseher. Die Alternative bestünde wohl darin, raus zu gehen und verkatert im Schnee herumzustapfen.

Was sie alle am Fernseher kleben lässt, ist ein einzigartiges Ereignis: der Ball des Präsidenten. In Finnland ist so etwas wie eine High Society weitgehend unbekannt, und deshalb ist der Ball des Präsidenten das einzige wirklich große gesellschaftliche Ereignis. Er wird landesweit ausgestrahlt, um die Leute davon abzuhalten, verkatert Auto zu fahren, und um zu beweisen, dass wir unsere eigene respektable Version der Oscar-Verleihung auf die Beine stellen können. Nein, es gibt noch eine bessere Metapher dafür: Der Ball ist die Super Bowl der finnischen High Society.

So kommt es, dass Finnen von Utsjoki im Norden bis Hanko im Süden einen ganzen Tag lang gebeizten Lachs und Aspirin in sich hineinstopfen und zuschauen, wie die Geladenen - Männer im Frack und Frauen in (für Skandinavien) gewagten Abendkleidern - am Präsidenten vorbeidefilieren und ihm die Hand drücken.

Neunzehnhundertneunundneunzig war das Jahr, in dem ich eingeladen wurde.

Du wirst automatisch eingeladen, wenn du finnischer Botschafter bist oder dem finnischen Parlament angehörst. Dazu kommen ungefähr einhundert oder zweihundert ausgewählte Gäste, die aus unterschiedlichsten Gründen eingeladen werden. Sie haben eine Goldmedaille bei den olympischen Spielen gewonnen oder vielleicht den Präsidenten im Wahlkampf unterstützt. Wenn du der Kapitän der Eishockeymannschaft bist und gerade die Weltmeisterschaft gewonnen hast, wirst du eingeladen. Wenn das von dir entwickelte Betriebssystem weltweite Beachtung findet, bist du dabei. Ehefrauen und Lebensgefährtinnen sind ebenfalls willkommen.

Tatsächlich war es reines Glück, dass Tove und ich überhaupt hingehen konnten. Im August hatten wir bei der amerikanischen Immigrationsbehörde die Genehmigung beantragt, nach Finnland zu reisen und danach wieder in die USA zurückzukehren. Wir bekamen die Genehmigung erst Anfang November. Zwei Wochen später erhielten wir die Einladung zum Ball des Präsidenten.

Man muss sich ausmalen, wie es dort zugeht. Stellen Sie sich 2000 Finnen vor - es brauchen noch nicht einmal die wichtigsten 2000 Finnen zu sein -, die sich in der Residenz des Präsidenten drängen. Sie war früher einmal das Anwesen eines russischen Geschäftsmanns gewesen. Eigentlich ist sie nur ein großes Haus, nicht gerade ein Einfamilienhaus, aber vielleicht ein Haus für eine Familie mit einer Menge Personal - Köche, Zimmermädchen und so. Jedenfalls ist sie nicht gerade riesig.

Du kommst also an. Jemand nimmt dir den Mantel ab und dann stehst du einfach eingezwängt in der Menge herum. Du weißt nicht, wohin. Es gibt reichlich Punsch. Offensichtlich ist Wodka darin. Du wärst nicht in Finnland, wenn es nicht so wäre. Es dauert eine Weile, bis du jemanden findest, mit dem du dich unterhalten kannst. Am Schluss redest du mit Journalisten, denn, ganz ehrlich, sie sind dort die interessantesten Leute. (Vielleicht lag es am Punsch, dass sie interessanter wirkten als ein Parlamentsabgeordneter aus, sagen wir, Lahti.)

Ich hatte nicht erwartet, besonders viel Spaß zu haben, weil ich nicht viele Leute dort kannte. Ich war der Einzige aus der Open-SourceSzene, der eine Einladung erhalten hatte. Ich dachte, es würde so ähnlich sein wie bei der Armee - das Nachtarocken hinterher wäre das Schönste daran. Aber dann machte es uns doch Spaß.

Tove trug ein aufregendes grünes Abendkleid, das die Aufmerksamkeit der Medien selbst dann erregt hätte, wenn wir der Oscar-Verleihung und nicht dem Ball des finnischen Präsidenten beigewohnt hätten. Weil sie so gut aussah und weil Finnland in diesem Jahr nicht die Eishockey-Weltmeisterschaft gewonnen hatte, ernannte uns die Presse zum König und zur Königin des Balls. Oder so.

„Sie betreten dieses Haus nicht als Journalist, sondern als Freund. In diesem Haus sind keine Journalisten zugelassen.“

Ich habe Tove noch nie so überschwänglich gesehen. Sie begrüßte mich an der Tür des neuen Hauses an dem Tag, an dem sie und Linus den Schlüssel bekommen hatten. Es ist eines jener Monsterhäuser: Der Medienraum (in dem jetzt Linus' Billardtisch steht) hat wahrscheinlich eine andere Postleitzahl als das Super-Bonus-Zimmer, in dem Patricia und Daniela schlafen, das seinerseits so groß ist, dass es einen ganzen Kindergarten beherbergen könnte. Von der Haustür führt ein breiter, endlos langer, verwinkelter Gang zum Familienwohnzimmer ganz hinten. Wenn sie die edlen italienischen Fliesen herausreißen, werden die Mädchen dort eines Tages ihre Skateboard-Künste erproben können. Linus' Arbeitszimmer im ersten Stock hat eine verspiegelte Schiebetür aus Glas. Fünf Badezimmer. Vielleicht haben sie in der Zwischenzeit noch ein paar mehr gefunden. Das Ganze liegt in einer geschlossenen, bewachten Wohnanlage weit entfernt vom Herzen des Silicon Valley.

Nicke Torvalds ist zu Besuch. Vater und Sohn kehren in einem gemieteten BMW Z3 von einem Abstecher in die alte Doppelhaushälfte zurück. Es ist das Modell, das Linus sich bald anschaffen wird, und Nicke wird am Nachmittag damit zur Bibliothek der Stanford University fahren. Vorerst aber lehnt er sich gegen den Hot-Tub in dem naturbelassenen Garten und verkündet, dies sei das größte Haus, das je ein Torvalds besessen habe. Dann listet er auf einem Blatt Papier alle zwanzig Torvalds auf. Er wusste noch nicht, dass eine Einundzwanzigste unterwegs war.

Auch Linus ist von dem noch leeren Haus hingerissen. Nicke nimmt die Umgebung auf Video auf, und ich bitte Linux, Tove über die Schwelle zu tragen, damit ich das Ereignis fotografieren kann. Das führt zu einem sehr un finnischen, öffentlichen Austausch von Zärtlichkeiten. „Hättest du jemals gedacht, dass unser Haus so groß sein würde?“ fragt mich Tove.

Tove wollte die Eröffnung der Ikea-Filiale in Emeryville nutzen, um Schränke für das neue Haus auszusuchen, und deshalb lud ich Linus ein, mich mit den Kindern am Stinson Beach zu besuchen. Gleich bei ihrer Ankunft drängte ich Linux, das Kajak in der Lagune auszuprobieren. Er paddelte zuerst alleine damit herum, dann mit jedem der Mädchen und kletterte schließlich mit nassen Hosen zurück auf den Bootssteg.

Ich wollte Linus' Meinung zu einem Kapitel mit dem Titel, „Wird mir der Erfolg zu Kopf steigen?“ hören und ging mit den Mädchen zum Strand hinunter, so dass er ungestört lesen konnte. Patricia und Daniela vergnügten sich ungefähr eine halbe Stunde lang damit, nach Seesternen zu suchen und ihre Zehen ins Wasser zu stecken. Dann verkündete eine von ihnen „Kisin kommer“, was übersetzt heißt: „Ich muss mal.“

Wir gingen ins Haus zu Linus zurück, der in Unterwäsche am Computer saß, eine Tüte Salzbrezeln neben sich, und vertieft vor sich hin tippte. Er brauchte ungefähr 15 Sekunden, um unsere Anwesenheit zu bemerken. Er sah vom Bildschirm hoch. Seine ersten Worte waren: „Mann, dein Mac ist zum Kotzen.“

Dann: „Ach ja, ich habe meine Hose in deinen Wäschetrockner gesteckt.“ Er hatte das Kapitel in „Ruhm und Reichtum“ umbenannt, weil er den Titel, „Wird mir der Erfolg zu Kopf steigen?“ etwas zu ichbezogen fand. Er wollte noch daran weiterarbeiten, deshalb ging ich mit den Mädchen hinaus, um nach Seehunden Ausschau zu halten, und ließ ihn das Kapitel zu Ende bringen.

MI

Es ist leicht, gegen Windmühlen zu kämpfen, wenn dir nicht bewusst ist, wie schwer es ist.

Wenn mich Leute vor fünf Jahren fragten, ob ich glaube, dass Linux eines Tages den Desktop-Bereich übernehmen und Microsoft in die Quere kommen könnte, schwang in ihrer Stimme immer ein leiser Zweifel mit. Ich antwortete unweigerlich, dass ich das sehr wohl glaube. Sie warfen mir einen skeptischen Blick zu. Tatsache ist, dass sie die Realität wahrscheinlich besser kannten als ich.

Ich habe mir all die Schritte, die dafür nötig sein würden, nie wirklich vor Augen geführt. Weder die Schritte, die zur Lösung der technischen Probleme bei der Entwicklung eines robusten und portablen Betriebssystems nötig sind, noch die Maßnahmen, die es braucht, um ein Betriebssystem nicht nur technisch, sondern auch kommerziell zum Erfolg zu führen. Ich wäre entmutigt gewesen, wenn ich im Voraus gewusst hätte, wie viel Infrastruktur erforderlich ist, um so erfolgreich zu werden, wie Linux es wurde. Es reicht nicht aus, gut zu sein. Natürlich musst du gut sein, aber außerdem müssen eine Menge anderer Dinge richtig laufen.

Jeder vernünftige Mensch hätte den schroffen Berg hochgeblickt, den es zu bezwingen galt, und wäre absolut verzagt gewesen. Denken Sie nur allein an die technischen Probleme, PCs zu unterstützen, die mehr Unterschiede zueinander aufweisen als jede andere Hardware auf der Welt. Du musst Leute unterstützen, bei denen in Anwendungen, die dich noch nicht einmal interessieren, Fehler auftreten, die du nicht reproduzieren kannst. Aber Linux liegt dir am Herzen und deshalb liegt dir daran, die Probleme zu lösen.

Um an ein Eindringen in den kommerziellen Markt überhaupt nur denken zu können, brauchst du ein beträchtliches Maß an Unterstützung für den Anwender. Linux wurde von Anfang an gut unterstützt, wenn es in Firmen eingesetzt wurde. Aber um wirklich groß in den Markt einzusteigen, brauchst du eine Menge Personal und eine zuverlässige Infrastruktur. Es reicht nicht aus, eine gebührenfreie Nummer für die ersten dreißig Tage zu haben. Bis zu einem gewissen Grad ist Anwenderunterstützung kein großes Thema mehr, weil du sie dir bei einer Reihe von Firmen kaufen kannst - Linuxcare, Red Hat, IBM, Silicon Graphics, Compaq, Dell. Aber sie muss eben gewährleistet sein. Die längste Zeit war mir das noch nicht einmal bewusst. Das war jahrelang eine große Herausforderung.

Im Gegensatz zu Betriebswirtschaftlern mit solidem technischen Hintergrund oder Journalisten mit wirtschaftlicher Ausrichtung war ich ein hochspezialisierter Softwareentwickler, der sich keine Vorstellungen machte, was dazu alles nötig war. Die technischen Probleme allein hätten mich davon abgehalten, das Wagnis einzugehen. Wenn ich gewusst hätte, wie viel Arbeit es bedeuten und dass ich zehn Jahre später immer noch damit beschäftigt sein würde - und zwar fast von morgens bis abends, zehn Jahre lang -, hätte ich das Ganze nie angefangen.

Und dann die Beschimpfungen! Okay, heute bekomme ich nicht mehr allzu viele Beschimpfungen zu hören. Aber ich muss damit rechnen. Leute, denen der Open-Source-Gedanke missfällt oder die sich über Fehler aufregen, senden mir E-Mails und laden ihren Frust bei mir ab. Der Anteil solcher E-Mails ist verschwindend gering, verglichen mit der Zahl der positiven Mails, die ich bekomme. Aber ich muss nach wie vor damit rechnen.

Ja, wenn ich erkannt hätte, wie viel Arbeit mich erwarten und als wie schwierig und mühevoll sich vieles erweisen würde, hätte ich das Ganze wahrscheinlich nie gemacht. Wenn ich das Wissen besessen hätte, die Probleme vorherzusehen, hätte ich Linux nach der ersten Freigabe wahrscheinlich nicht sehr viel weiter vorangetrieben.

Wenn ich gewusst hätte, wie viele Details du richtig hinbekommen musst und welche Erwartungen an ein Betriebssystem gestellt werden, hätte ich mich mit Horrorszenarien von unlösbaren Problemen herumgeplagt.

Okay. Andererseits konnte ich auch die Pluspunkte nicht vorhersehen. Zum Beispiel wie viel Unterstützung ich haben würde und wie viele

Leute bereit waren, gemeinsam an der Sache zu arbeiten. Deshalb will ich mich an dieser Stelle berichtigen. Wenn ich die Pluspunkte gekannt hätte, hätte ich es wahrscheinlich *doch* gemacht.

Geistiges Eigentum

Die Frage nach dem geistigen Eigentum wird in diesen Tagen so heiß diskutiert, dass ich keine Toilette betreten kann, ohne auf ein Graffiti zu stoßen, das die eine oder die andere Seite unterstützt. Manche Leute meinen, Patente und andere Formen zum rechtlichen Schutz des geistigen Eigentums seien der Ruin des freien Universums, und die entsprechenden Gesetze seien nicht nur fehlgeleitet, sondern *böse* und sollten so bald wie möglich abgeschafft werden. Andere sind davon überzeugt, das geistige Eigentum sei der Antriebsmotor der ganzen Weltwirtschaft. Und diese Leute wollen alles dafür tun, die Gesetze zum Schutz des geistigen Eigentums zu verschärfen.

Infolgedessen sind die Graffitis darüber manchmal recht anschaulich.

Natürlich sind die meisten davon in den virtuellen Toiletten des Internets zu finden, nicht so sehr in den Toiletten der Szenelokale des Nachtlebens von San Jose." Dort werden gewaltige Dispute über spezielle Details der Gesetze zum geistigen Eigentum ausgetragen. Die Leute streiten sich über alles und jedes, vom ersten Zusatzartikel der amerikanischen Verfassung bis hin zu der Frage, ob die Gesetze zum geistigen Eigentum Open-Source-Entwicklungen in der Zukunft unmöglich machen.

Und leider stelle ich fest, dass ich selbst diesem Thema derart schizophren gegenüber stehe, dass man mich eigentlich einweisen müsste.

Nicht, dass ich keine Meinung dazu hätte: Ich habe sehr ausgeprägte Meinungen über den Wert des geistigen Eigentums, aber sie unterstützen beide Seiten des Streits. Ich kann Ihnen sagen, das kann sehr verwirrend sein. Weil ich nämlich gleichzeitig für beide Seiten eintrete.

Vermutlich liegt das daran, dass geistiges Eigentum wirklich zwei Seiten hat, die nichts als den Namen gemeinsam haben.

Für viele Leute, unter anderem für mich, hat geistiges Eigentum mit menschlichem Erfindungsgeist zu tun, mit dem, was uns zu Menschen macht und uns von Tieren unterscheidet. Vor diesem Hintergrund ist schon der Name „geistiges Eigentum“ ein Affront: Geistiges Eigentum ist nicht ein „Eigentum“, das wie ein bewegliches Gut verkauft werden kann, es ist ein Schöpfungsakt, das Größte, was ein Mensch je leisten kann. Es ist KUNST, in Großbuchstaben geschrieben. Es kann die Mona Lisa sein, aber es kann auch das Endergebnis einer langen Programmernacht sein, und es ist ein Endergebnis, auf das du als Programmierer verdammt stolz bist. Es ist so wertvoll, dass es nicht verkauft werden kann: Es ist unauslöschlich ein Teil dessen, was du bist.

Diese Art von Kreativität - ganz gleich, ob sie sich in Form eines Gemäldes, eines Musikstücks, einer Skulptur, eines Textes oder als Programmcode ausdrückt - sollte heilig sein: Der Schöpfer und seine Schöpfung sind durch ein untrennbares Band miteinander verbunden. Es ist wie das Band zwischen einer Mutter und ihrem Kind, oder zwischen schlechtem chinesischem Essen und Glutamat. Gleichzeitig ist es etwas, woran jeder in der Welt Anteil haben sollte, denn es ist etwas zutiefst Menschliches.

Auf der anderen Seite aber ist geistiges Eigentum, dessen Wert sich auf etwa sieben Zillionen Milliarden U.S.-Dollar im Jahr beziffern lässt, ein Riesengeschäft. Die menschliche Kreativität wurde mit einem Preisschild versehen und erwies sich als ziemlich teuer. Kreativität ist selten und deshalb nicht nur teuer, sondern auch extrem lukrativ. Und damit kommt eine ganz andere Kategorie von Argumenten ins Spiel, und eine ganz andere Art von Leuten tritt auf den Plan. Die Art von Leuten, die das Endergebnis der menschlichen Kreativität als „Eigentum“ bezeichnen. Und natürlich Anwälte.

Schauen Sie sich die Kapitelüberschrift noch einmal an. Die „Eigentumsleute“ sind dabei, zu gewinnen. Ihr Name hat sich durchgesetzt. Wo liegt also das Problem?

Das bekannteste Beispiel für geistiges Eigentum ist das Konzept des Urheberrechts. Urheberrechte sind im Prinzip nichts anderes als die Festschreibung des Rechts jedes Schöpfers, mit seiner Schöpfung zu tun, was er möchte. Der „Eigentümer“ der Schöpfung kann entscheiden, wie seine Schöpfung verwendet werden darf.

Urheberrechte lassen sich rechtlich sehr einfach beschaffen. Sie müssen Ihr Urheberrecht nicht eintragen lassen: Sie haben das Urheberrecht Ihrer kreativen Arbeit automatisch inne. Das ist ein wichtiger Unterschied zu den meisten anderen Möglichkeiten zum Schutz des geistigen Eigentums, vor allem, weil auf diese Weise auch Einzelpersonen Urheberrechte besitzen können, nicht nur große Unternehmen. Sie können ein Urheberrecht besitzen, weil Sie etwas geschrieben, gemalt oder einfach ganz allgemein etwas Einzigartiges geschaffen haben. Wenn Sie wollen, können Sie einen Vermerk wie „© Copyright 2000 Ihr Name“ anbringen, aber das ist ehrlich gesagt gar nicht nötig. Sie besitzen das Urheberrecht, ganz gleich, ob sie darauf verweisen oder nicht. Der Verweis macht es anderen Leuten nur leichter, Sie aufzuspüren, wenn sie Ihre Arbeit nutzen möchten.

Der Besitz eines Urheberrechts für sich allein ist natürlich nicht sonderlich nützlich. Aber die Tatsache, dass Sie besitzen, was Sie geschaffen haben, bedeutet, dass Sie die Nutzung Ihres Werks steuern können. Sie haben zum Beispiel das Recht, Ihr Kunstwerk an einen anderen zu verkaufen, und niemand außer dem Finanzamt hat dabei mitzureden. Aber es geht um mehr als nur um Geld, und dieser Punkt scheint viele Leute aus der Fassung zu bringen.

Beispielsweise können Sie Ihre Macht als Eigentümer des Urheberrechts für etwas Interessanteres nutzen als für einen Verkauf Ihres Werks. Sie können es lizenzieren. Das ist sogar noch besser, als es zu verkaufen. Statt das Kunstwerk zu verkaufen, können Sie die Lizenz verkaufen, etwas Bestimmtes damit anzufangen, und dennoch das Urheberrecht behalten. Im Prinzip können Sie den Kuchen haben und essen. So entstehen die Microsofts dieser Welt: Sie verkaufen endlos die Rechte, etwas zu verwenden, ohne selbst etwas dabei zu verlieren. Kein Wunder, dass jeder darauf *scharf* ist, diese Art von Eigentum zu besitzen.

Dämmert jetzt vielleicht jemandem, wo das Problem liegt? Wenn Ihnen immer noch nichts auffällt, habe ich eine Brücke und ein paar Gebäude an der Uferpromenade, die ich Ihnen verkaufen kann.

Allmählich nähern wir uns dem Grundproblem des geistigen Eigentums: Sie als Eigentümer des geistigen Eigentums können es *ad infinitum* verkaufen, ohne selbst etwas dabei zu verlieren. Sie riskieren nichts, und Sie könnten Ihre Lizenz sogar so abfassen, dass man Sie, sollte sich das Eigentum als fehlerhaft erweisen, in keiner Weise zur

Verantwortung ziehen kann. Das klingt absurd? Sie wären überrascht. Das Problem dabei: kein Verbraucherschutz.

Aber es kommt noch schlimmer. Der Eigentümer des Urheberrechts hat nicht nur das Recht, sein Eigentum zu verkaufen, ohne es zu verlieren, er hat auch das Recht, andere Menschen gerichtlich zu belangen, die Eigentum verkaufen, das wie sein Eigentum *aussieht*. Offensichtlich hat der Eigentümer des Urheberrechts Rechte über die von seiner Arbeit *abgeleitete Arbeit*.

Offensichtlich? Nicht so schnell. Wo ziehen Sie die Linie zwischen Inspiration und Imitation? Und was passiert, wenn unterschiedliche Leute auf eine ähnliche Idee gekommen sind? Wer bekommt dann den Blankoscheck, seine Idee immer und immer und immer wieder zu verkaufen und von allen anderen zu verlangen, sich aus dem Geschäft heraus zu halten? Das Konzept des „geistigen Eigentums“ benachteiligt nicht nur die Verbraucher, sondern auch *andere kreative Menschen*.

An diesem Punkt wird die Diskussion hässlich, weil viele der Argumente für stärkere Gesetze zum Schutz des geistigen Eigentums auf dem Gedanken basieren, Erfinder und Künstler besser abzusichern. Dabei fällt manchen Leuten noch nicht einmal auf, dass die Verleihung derart mächtiger Rechte an die einen die Rechte anderer beschneidet.

Es ist vielleicht nicht sonderlich überraschend, dass die eifrigsten Befürworter strengerer Gesetze zum Schutz des geistigen Eigentums die Organisationen sind, die am meisten zu gewinnen haben. Nicht die Künstler und Erfinder selbst, sondern die Verwerter des geistigen Eigentums: Unternehmen, die ihren Lebensunterhalt mit der Kreativität anderer Leute verdienen. Ach ja, und die Anwälte, nicht zu vergessen. Das Ende vom Lied? Ergänzungen zum Urheberrecht wie das berühmte Digital-Millennium-Copyright-Gesetz, das Verbraucher, die urheberrechtlich geschütztes Material einsetzen, noch ihrer letzten Rechte beraubt.

Falls Sie jetzt den Eindruck gewonnen haben sollten, ich hielte Urheberrechte für schädlich, liegen Sie falsch. Zufällig bin ich ein begeisterter Anhänger von Urheberrechten, ich halte nur nichts von einer Überbetonung der Autorenrechte. Urheberrechte enden da, wo Verbraucher über den Tisch gezogen werden. Und ich sage das nicht nur

als Verbraucher, sondern ausdrücklich auch als Produzent von urheberrechtlich geschütztem Material: in der Form dieses Buches und in der Form von Linux selbst.

Ich als Eigentümer des Urheberrechts habe meine Rechte. Aber Rechte bringen Pflichten mit sich oder, wie es in manchen Kreisen heißt, *Noblesse oblige*. Und daher ist es meine Pflicht, meine Rechte verantwortlich zu nutzen und nicht als Waffe gegen andere einzusetzen, die keine vergleichbaren Rechte besitzen. Wie ein großer Amerikaner einmal sagte: „Fragen Sie nicht, was Ihr Urheberrecht für Sie tun kann, fragen Sie, was Sie für Ihr Urheberrecht tun können“ - oder so ähnlich.

Und unterm Strich betrachtet ist das Urheberrecht ungeachtet des Digital-Millennium-Copyright-Gesetzes durchaus eine relativ milde und gemäßigte Form, geistiges Eigentum zu schützen. Ihm wohnt noch der Gedanken des „fairen Gebrauchs“ inne, und es räumt dem Eigentümer des Urheberrechts nicht *alle* Rechte an der Arbeit ein.

Das Gleiche kann man von Patenten, Warenzeichen und Geschäftsgeheimnissen nicht behaupten: den schweren Drogen des geistigen Eigentums.

Insbesondere Softwarepatente werden in technischen Kreisen mittlerweile so erbittert diskutiert, dass sie dem offiziellen Kanon der Themen angehören, über die man in Gesellschaft genauso wenig spricht wie über Waffen- und Abtreibungsgesetze, den Einsatz von Marihuana für medizinische Zwecke und die Frage, ob Pepsi Cola besser schmeckt als Coca Cola. Das liegt daran, dass Patente ihren Inhabern in vieler Hinsicht eine ähnliche Kontrolle über neue Erfindungen verleihen wie Urheberrechte, ohne deren ausgleichendes Moment zu enthalten.

Patente haben unter anderem den großen Nachteil, dass du mit einer Neuerfindung nicht automatisch auch das Patent dafür bekommst. Nein, du bekommst ein Patent erst, nachdem du dich dem mühsamen und langwierigen Prozess unterzogen hast, es beim Patentamt anzumelden. Beim Patentamt zu warten, ist übrigens ein bisschen, als würdest du bei der Kfz-Zulassungsstelle in der Schlange stehen, nur mit dem Unterschied, dass du ungefähr zwölf Anwälte im Schlepptau hast und dass die Warteschlange ZWEI JAHRE LANG ist. Kurz gesagt, ein Patent meldet man nicht einfach so zum Spaß an einem Freitagabend an, an dem die Kinder mal früher als sonst eingeschlafen sind.

Um dem Ganzen die Krone aufzusetzen, verfügt das Patentamt nicht unbedingt über die Ressourcen, die Außerordentlichkeit deiner Neuerung zu prüfen. Schließlich arbeiten dort keine Einsteins^{*}, und deshalb ist eine entsprechende Überprüfung neuer Erfindungen ziemlich schwierig. Was bedeutet, dass viele Patente zu Unrecht akzeptiert wurden. Sie müssen es sich wie ein Postamt mit promovierten Mitarbeitern vorstellen - abzüglich der Leute vom Sicherheitsdienst.

Was schließen wir daraus? Aus offensichtlichen Gründen werden Patente nur selten von Einzelpersonen erworben. *Unternehmen* dagegen erwerben sie *en gros*. Sie eignen sich gut als Waffen gegen andere Unternehmen, die mit Prozessen wegen der in *ihrem* Besitz befindlichen Patente drohen. Das Patentsystem von heute ist im Prinzip ein Kalter Krieg, der statt mit Kernwaffen mit geistigem Eigentum geführt wird. Und er verläuft kaum friedvoller. Die Leute, die zusammengekauert in den Bombenbunkern zurückbleiben, sind die einzelnen Erfinder, die mit einem verrückt gewordenen System zurechtkommen müssen, und denen die Mittel fehlen, 12000 Anwälte zu beschäftigen.

Okay, wenn du den Stress mit Patenten scheust, kannst du zur stärksten Droge beim Schutz von geistigem Eigentum greifen: zum Geschäftsgeheimnis. Geschäftsgeheimnisse haben den Vorteil, dass du dich nicht mit so einer Art Geschäftsgeheimnisamt herumzuschlagen brauchst. Du erklärst dein geistiges Eigentum einfach zum Geschäftsgeheimnis und fertig. Du kannst anderen davon erzählen, musst ihnen aber sagen, dass es ein Geheimnis ist.

Dieses Vorgehen war früher allgemein üblich, und genau aus diesem Grund wurde das Patentgesetz ursprünglich eingeführt. Um Einzelpersonen und Unternehmen zur Preisgabe ihrer Geheimnisse zu ermutigen, machte das Patentgesetz es möglich, eine Erfindung eine gewisse Zeit lang am Markt zu schützen, *wenn man im Gegenzug sein Erfolgsgeheimnis enthüllte*. Es handelte sich dabei um eine einfache Form des Gebens und Nehmens: Du sagst uns, wie du etwas machst, und wir überlassen dir dafür die Exklusivrechte für X Jahre.

Vor der Einführung von Patenten hüteten die Leute argwöhnisch ihren technologischen Wissensvorsprung und nahmen ihre Geheimnisse mit

ins Grab. Das war aus offensichtlichen Gründen schlecht für die technologische Entwicklung, weil viel versprechende Techniken keinem Fremden preisgegeben wurden. Als dann mit der Einführung von Patenten Exklusivrechte in Aussicht standen, war der Anreiz groß, das eigene Wissen offen zu legen: Du musstest dir keine Sorgen mehr machen, die Konkurrenz könnte dir auf die Schliche kommen und dein Geheimnis seinen Wert verlieren.

Aber das war damals, und heute ist heute. Heutzutage genießen aus unerfindlichen Gründen sogar Geschäftsgeheimnisse einen gesetzlichen Schutz. Jedem vernünftigen Menschen ist klar, dass ein Geheimnis, das aufgefliegen ist, kein Geheimnis mehr ist. Nur die seltsamen, verworrenen Passagen der Gesetze zum geistigen Eigentum sehen das anders. Ihnen zufolge können Geschäftsgeheimnisse bleiben, auch wenn jeder sie längst kennt. Und ihnen zufolge kann jeder rechtliche Schritte gegen das Wissen in deinem Kopf einleiten, wenn du zufällig für den falschen Arbeitgeber tätig bist. Ein Teil der Gesetze zum Schutz des geistigen Eigentums ist wirklich zum Fürchten.

Es ist ein vorrangiges Anliegen von Open Source, den Krieg um das geistige Eigentum zu beenden. Auch wenn die Meinungen über den Sinn des Open-Source-Gedankens geteilt sind, kann man ihn in vieler Hinsicht als Beitrag zur Entspannung im Hightech-Bereich betrachten: Er entschärft die Waffe, die das Urheberrecht im Kampf um das geistige Eigentum darstellt.

Deshalb würde die Open-Source-Bewegung die gesetzlich zulässige Waffe des Urheberrechts gerne als Einladung zu einer gemeinsamen Party betrachten statt als Waffe, die sich gegen andere richtet. Nach wie vor gilt das alte Mantra: *Make Love, Not War*, nur auf einer etwas abstrakteren Ebene (vielleicht einer *viel* abstrakteren, wenn man sich einige der Geeks ansieht, die ich kenne).

Aber jede große philosophische Auseinandersetzung hat ihre Kehrseite. An diesem Punkt kommt meine schwere Schizophrenie ins Spiel.

Ich habe versucht zu erklären, warum viele Leute geistiges Eigentum und insbesondere die Stärkung der Gesetze zum Schutz des geistigen Eigentums für eine schlimme Sache halten. Viele innerhalb (und ehrlich gesagt auch außerhalb) der Open-Source-Szene würden Kernwaffen am liebsten komplett abschaffen und den Kalten Krieg des Wissens ein für alle Mal beenden. Andere sind dagegen.

* Einstein arbeitete wirklich im Patentamt, als er die Relativitätstheorie

Andererseits ist der Schutz des geistigen Eigentums womöglich wirklich unfair geregelt, und die entsprechenden Gesetze sind primär dazu angetan, die Ziele großer Unternehmen über die Rechte der Verbraucher oder sogar des einzelnen Autors oder Erfinders zu stellen. Aber, Mann, Urheberrechte sind lukrativ. Sie verstärken die Macht der Mächtigen, und gerade die Tatsache, dass sie eine so starke Waffe sind, verleiht ihnen ihre Wirksamkeit. Aus dem gleichen Grund, aus dem Kernwaffen zur ultimativen Waffe des Kalten Krieges wurden, wird das Recht zum Schutz des geistigen Eigentums zu einer attraktiven Waffe im Krieg der Technologie. Und Technologie verkauft sich.

Darüber hinaus ist das Urheberrecht in hohem Maße selbstverstärkend. Weil geistiges Eigentum eine so hervorragende Einnahmequelle darstellt, wird eine Menge Geld investiert, um *mehr* geistiges Eigentum zu schaffen. Diese Tatsache ist wichtig. Genauso wie Kriege geschichtlich betrachtet immer eine Quelle neuer Erfindungen und großer technologischer Fortschritte waren (der Computer selbst wurde anfangs vor allem für rein militärische Zwecke entwickelt), trägt der virtuelle Kampf um das geistige Eigentum dazu bei, den Motor anzukurbeln, und bisher ungeahnte Ressourcen in die technische Entwicklung zu pumpen. Das ist eine gute Sache.

Natürlich bin ich als intellektueller Snob davon überzeugt, dass das bloße Herumwerfen mit Ressourcen wahre Kreativität nicht sonderlich fördert. Denken Sie nur an die Musikbranche. Zillionen von Dollars werden jedes Jahr verschleudert, um den nächsten kippen Künstler zu finden - und doch denkt niemand wirklich, die Spice Girls (die für ihren Beitrag zu ihrer Kunstform reich entlohnt wurden) könnten sich mit Wolfgang Amadeus Mozart messen (der mittellos starb). Anscheinend bringt das Herumwerfen mit Geld nicht die größten Genies hervor.

Aber intellektueller Snobismus - die Genie-lässt-sich-mit-Geld-nichtkaufen-Philosophie - ist als langfristiges Geschäftsmodell ungeeignet. Die kreativen Säfte sind so unvorhersehbar, sie lassen sich so schwer zuverlässig aus dem Hut zaubern, dass eine langfristige Planung sich nicht in der Hoffnung auf einen Geniestreich erschöpfen sollte. Die Technologieentwicklung (und traurigerweise auch die Musik) von heute hängt nicht von den Einsteins und Mozarts ab, sondern von

nur gelegentlich aufblitzt. Die zusätzlichen Ressourcen sorgen nicht für große Kunst, sondern für ein langsames, stetiges Voranschreiten. Doch am Ende wird alles gut.

Das Konzept der schuftenden Ingenieure mag weniger romantisch sein als das Bild des exzentrischen Genies. Denken Sie nur daran, wie viele Filme es über verrückte Wissenschaftler gibt verglichen zur Zahl der Filme über schuftende Ingenieure. Aber wenn es ums Geschäft geht, brauchen Sie den stetigen Fluss der kleinen Verbesserungen, die sich im Lauf der Zeit ergeben, noch dringender als das gelegentliche Aufblitzen von Genie.

Und hier zeigt sich die ganze Macht des geistigen Eigentums: Weil es so lukrativ ist, wurde es zum heiligen Gral moderner Hightech-Firmen, die das System mit immer neuem Stoff versorgen. Und dank der Gesetze zum Schutz des geistigen Eigentums nimmt der stetige Fluss

ungehindert seinen Lauf.
aber er ist verlässlich.

Er mag nicht mehr sonderlich kreativ sein,

Ich sehe also beide Seiten. Obwohl ich zugeben muss, dass ich meistens lieber eine amüsantere und inspirierendere Welt der Technik vor mir sehen würde. Eine, in der nicht *immer* die wirtschaftlichen Faktoren Vorrang haben. Ich habe einen Traum - eines Tages werden die Gesetze zum Schutz des geistigen Eigentums von der Moral geleitet sein, nicht von der Frage, wer das größte Stück des Kuchens bekommt.

Glauben Sie mir, ich verstehe, was wirtschaftlich auf dem Spiel steht. Gleichzeitig kann ich nicht umhin zu wünschen, wirtschaftliche Überlegungen hätten keinen so überwältigend negativen Einfluss auf die modernen Gesetze zum Schutz des geistigen Eigentums. Das wirtschaftliche Interesse, geistiges Eigentum noch besser zu schützen, und die Schwierigkeit, die Konzepte „fairer Gebrauch“ und „Moral“ in Gesetzestexten auszudrücken, ließen die gegensätzlichen Standpunkte zur Frage des geistigen Eigentums noch weiter auseinander rücken. Wie bei einem Nachbarschaftsstreit will keine der beiden Seiten zugeben, dass die richtige Lösung wahrscheinlich irgendwo in der Mitte zwischen den beiden Extremen liegt.

Natürlich spielen, wie der unglückliche Erlass des Digital-Millennium-Copyright-Gesetzes zeigte, finanzielle Anreize eine wichtige Rolle. Die Frage ist nur: Welche Art von Gesetz zum Schutz des geistigen Eigentums kann den Fortschritt befördern und gleichzeitig weniger offensichtlich von schierer Geldgier motiviert sein?

Das Problem wird zusätzlich durch die Tatsache verstärkt, dass die moderne Technologie (und insbesondere das Internet) viele der traditionellen Formen zum Schutz des geistigen Eigentums schneller schwächen, als wir Schritt halten können. Und auf eine Art, die niemand vorhersehen konnte. Ich meine - Mann, wer hätte sich vorstellen können, dass Großmütter aus dem Mittleren Westen Raubkopien von Stickanleitungen über das Internet verteilen würden? Die Möglichkeiten, Kunstwerke - und Techniken - im großen Stil zu kopieren, sind so breit gefächert und leicht zugänglich, dass Institutionen mit maßgeblichen Eigentumsrechten all ihren Einfluss zur Wahrung ihrer Interessen geltend machen. Sie tun alles in ihrer Macht stehende, um diese Art des Kopierens und sogar die Techniken, die für Raubkopien eingesetzt werden, für ungesetzlich zu erklären.

Was gibt es daran auszusetzen? Das Problem liegt auf der Hand: Viele der neuen Versuche, die ungesetzliche Verwendung von geistigem Eigentum durch Dritte zu erschweren, erschweren es gleichzeitig, die Arbeit anderer Leute auf legale Weise zu nutzen. In der Linux-Welt ist das klassische Beispiel dafür der so genannte DeCSS-Prozess.

Bei dem DeCSS-Prozess wurden Leute, die an einer Technologie zur Entschlüsselung von DVD-Filmen arbeiteten, von der Unterhaltungsindustrie verklagt, weil sie den Code anderen im Internet zur Verfügung stellten. Es spielte für den Richter keine Rolle, dass das eigentliche Ziel des Projekts völlig legal war: Nur weil das Projekt *potenziell* für illegale Zwecke eingesetzt werden konnte, war in den USA sogar die Verbreitung der Information verboten, wo die Entschlüsselungsbefehle zu finden waren. (Der Name „DeCSS“ ist vom Namen des Verschlüsselungsverfahrens abgeleitet, das für die DVD verwendet wird: Content Scrambling System - CSS. „De-CSS“ bedeutet somit, dass der CSS-Kopierschutz entfernt wird, um den Film auf dem Computer abspielen zu können.)

Das ist ein ausgezeichnetes Beispiel für einen Fall, in dem die Schutzrechte nicht zur Förderung von Innovationen, sondern zur Kontrolle des Marktes und zur Gängelung der Verbraucher genutzt werden. Es ist ein Beispiel für einen Fall, in dem der Schutz des geistigen Eigentums überzogen wurde.

Der Schutz des geistigen Eigentums kann übrigens nicht nur im Bereich der Technik zu Missbrauch führen. In einem anderen klassischen Beispiel wurde das Geschäftsgeheimnis herangezogen, um die

Leute zu belangen und unter Druck zu setzen, die versuchten, die Öffentlichkeit über Scientology zu informieren. Scientology trug erfolgreich vor, ihre Schriften („Advanced Technology“) unterlägen dem Schutz des Geschäftsgeheimnisses, und machten sich so die Gesetze zum Schutz des geistigen Eigentums zunutze, um eine Veröffentlichung der Informationen zu verhindern.

Was sind die Alternativen? Stellen Sie sich ein Gesetz zum Schutz des geistigen Eigentums vor, das auch die Rechte *anderer* Leute mit berücksichtigt. Stellen Sie sich Gesetze zum Schutz des geistigen Eigentums vor, die Offenheit und gegenseitigen Austausch ermutigen. Natürlich können Sie Ihre technischen oder religiösen Geheimnisse auch weiterhin für sich behalten, aber diese Geheimnistuerei sollte nicht unter dem Schutz des Gesetzes stehen.

Ja, ich weiß. Wie kann man nur so unrealistisch sein.

Nieder mit der Gängelei

Das Geheimnis des Überlebens und Florierens besteht darin, das bestmögliche Produkt herzustellen, das du zustande bringst. Und wenn du damit nicht überleben und florieren kannst, geschieht es dir recht. Wenn du kein gutes Auto bauen kannst, verdienst du es, unterzugehen wie die amerikanische Autoindustrie in den 1970er-Jahren. Erfolg hat mit Qualität zu tun und mit der Fähigkeit, den Leuten zu geben, was sie wollen.

Er hat nichts damit zu tun, andere zu gängeln.

Leider sind Menschen und Unternehmen zu oft von reiner Gier motiviert. Das führt dazu, dass sie auf lange Sicht verlieren. Gier bewirkt Entscheidungen, die von Paranoia und dem Bedürfnis nach totaler Kontrolle geleitet sind. Solche Entscheidungen sind schlecht und kurzsichtig und enden in der Katastrophe oder Beinahe-Katastrophe. Ein einfaches Beispiel dafür, das jeder kennt, war der phänomenale Durchbruch der drahtlosen Telekommunikation in Europa zu Lasten amerikanischer Firmen. Während die amerikanischen Unternehmen versuchten, durch Einsatz ihrer individuellen Markenstandards den Markt im Alleingang zu kontrollieren, einigten sich die europäischen Firmen auf einen gemeinsamen Standard, GSM, und konkurrierten darum, das beste Produkt und den besten Service zu liefern. Die amerikanischen Firmen sind seither ins Hintertreffen geraten und leiden unter ihren selbst gewählten konkurrierenden Standards. In einem Markt, dem ein gemeinsamer Standard Auftrieb gab, nahmen alle europäischen Firmen an dem Boom teil. Deshalb tauschten die Kids in Prag längst SMS-Botschaften aus, bevor die Kids in Peoria überhaupt von dieser neuen Möglichkeit des Spickens bei Prüfungen gehört hatten.

Wer versucht, Geld zu machen, indem er eine Ressource kontrolliert, fliegt über kurz oder lang aus dem Geschäft. In der Geschichte finden sich unzählige Beispiele, die die fatalen Auswirkungen dieser Form des

Despotismus illustrieren. Nehmen wir an, wir befinden uns um 1800 herum irgendwo im amerikanischen Westen, und Sie kontrollieren die Wasserquelle der örtlichen Farmer. Sie knausern mit dem Wasser und verlangen überhöhte Preise. Irgendwann wird unweigerlich ein anderer ein einträgliches Geschäft darin sehen, das Wasser von anderswo heranzuschaffen, und Ihr Markt bricht zusammen. Vielleicht ermöglicht es der technische Fortschritt, Wasser über Rohrleitungen aus größeren Entfernungen herzuleiten, oder die Umstände verändern sich. So oder so, Ihre Machtposition ist gebrochen, und Sie stehen vor dem Nichts. Das kommt ständig vor, aber erstaunlicherweise scheint niemand damit zu rechnen.

Kehren wir zurück in die Gegenwart und schauen wir uns die Musikbranche in den letzten Jahren des zu Ende gegangenen 20. Jahrhunderts an. Sie kontrolliert die Ressource Unterhaltung. Ein Unternehmen besitzt die Rechte an der Arbeit eines Künstlers. Dieser Künstler produziert eine Reihe erfolgreicher Titel. Aber auf den CDs, die das Unternehmen auf den Markt bringt, sind jeweils nur ein oder zwei dieser Titel enthalten. Auf diese Weise kann das Unternehmen mehrere CDs verkaufen, statt der einen, die jeder gerne hätte. Dann erfindet jemand die MP3-Technik. Plötzlich lassen sich Musiktitel vom Internet herunterladen. MP3 tut das Richtige: Es kommt den Kunden entgegen und eröffnet ihnen Wahlmöglichkeiten.

Wenn eine CD typischerweise 10 Dollar kostet, und nur zwei Titel enthält, die der Kunde will, ist es für ihn vielleicht sinnvoller, diese und andere Titel, an denen er wirklich interessiert ist, einzeln per MP3 für 1,50 Dollar pro Stück zu kaufen. Der Käufer braucht sich nicht mehr mit der despotischen Herrschaft und der raffgierigen Geschäftspolitik der Musikbranche abzufinden, die nur die Häppchen herausrückt, die sie herausrücken möchte. Es gibt einen guten Grund, warum der MP3-Standard und seine Schwestertechnologien Napster und Gnutella die Musikindustrie zu Tode erschrecken. Der Wasserpreis wurde so hoch, dass es sich für einen anderen lohnte, eine Methode zu ersinnen, anderweitig Wasser zu beschaffen.

In der Musikbranche aber hat der Versuch, die Kunden zu gängeln, eine lange Geschichte. Wenn nicht über die Auswahl der veröffentlichten Musiktitel, dann über Urheberrecht und Technologie. So versuchte die Branche, als in den 1960er-Jahren das Tonband aufkam, die Kunden mit allen Mitteln davon abzuhalten, Musiktitel zu kopieren. Sie

erkannte das Tonband als perfektes Medium, die Leute zum Unterlaufen der Urheberrechte zu animieren, und wollte ihre Copyrights besser geschützt wissen. Das war eine lahme Entschuldigung. Die Branche berief sich auf moralische Grundsätze und Urheberrechte, obwohl es ihr eigentlich nur darum ging, weiterhin ungestört in ihrer Nische schalten und walten zu können. Tatsache ist, dass Tonbänder der Musikbranche nie geschadet haben. Natürlich kopierten die Leute Musiktitel für den Eigengebrauch, letztendlich aber kauften sie mehr LPs als vorher, die sie dann kopieren konnten. Tja. Ein paar Jahrzehnte später, als die CD auf den Markt kam, waren die CD-Player so gebaut, dass Kopien auf Band nicht die gleiche Wiedergabequalität hatten. Wieder siegte die Paranoia. Als Nächstes kamen digitale Bänder. Sie hatten eine andere Abtastrate als CDs - 48 Kilohertz gegenüber 44,1 -, die verhindern sollte, dass die Anwender ihre CDs auf Digitalband überspielten. Um nur ja keine Kontrolle abgeben zu müssen, werden die Kunden gnadenlos über den Tisch gezogen. Übrigens waren die Digitalbänder nie ein sonderlicher Markterfolg. Mutter Natur lässt sich eben nicht ins Handwerk pfuschen.

Durch ihren Versuch, jede Nachfolgetechnik zu kontrollieren, trägt die Branche nur dazu bei, dass die Leute nach neuen Auswegen suchen. Ob sie es wohl je begreifen wird?

Damit kommen wir zum Thema DVDs. Dieses Mal brachte die Branche eine sehr viel bessere Ton- und Bildqualität zuwege, als wir sie von VHS-Bändern kennen. Zudem sind DVDs kleinformatiger und leichter zu bedienen. Aber eine Verschlüsselungstechnik verhinderte das Kopieren. Und um dem Ganzen die Krone aufzusetzen, waren die DVDs mit einem Regionalcode ausgestattet. Auf diese Weise war es unmöglich, eine am Flughafen von San Francisco gekaufte DVD in Europa abzuspielen. Für die Branche war das auf perverse Weise sinnvoll: Hey, Leute, wir können Filme in Europa teurer absetzen! Lasst uns deshalb dafür sorgen, dass Europäern die Lust vergeht, Filme in den USA zu kaufen.

Hätte die Unterhaltungsbranche die offensichtlichen Folgen nicht vorhersehen können? Dass der Wasserpreis so ansteigen würde, dass jemand eine neue Methode ersinnen würde, das kostbare Nass von anderswo herzuleiten?

Noch während die Branche in ihrer Gier versuchte, die Leute durch technische Schikanen zu kontrollieren, wurde die DVD-Verschlüsse-

Jung geknackt - und zwar ironischerweise nicht von Leuten, die DVDs kopieren wollten, sondern von Leuten, die sie einfach unter Linux ansehen wollten. Diese Leute wollten die DVDs tatsächlich *kaufen*, konnten es aber nicht, weil sie in ihrer Umgebung mit den Datenträgern nichts anfangen konnten. Die Winkelzüge der Branche, ihre Pfründe zu sichern, erwiesen sich als Eigentor: Sie verhinderten lediglich das Marktwachstum und schufen den Anreiz, die DVD-Verschlüsselung zu knacken. Wieder einmal erwies sich die kurzfristige Strategie als das Falsche.

Die Unterhaltungsindustrie ist nur ein Beispiel. Im Softwarebereich passiert seit Jahren genau das Gleiche. Deshalb ist die Microsoft-Strategie der Bündelung von Software letztendlich zum Scheitern verurteilt. Open-Source-Produkte dagegen können von vornherein nicht despotisch eingesetzt werden, weil sie frei zugänglich sind. Wenn jemand versuchen würde, etwas mit Linux zu bündeln, würde ein anderer die Bündelung einfach rückgängig machen und Linux in der Form verkaufen, die die Leute wirklich wollen.

Der Versuch, Menschen mit technischen Mitteln zu gängeln, ist gleich in zweifacher Hinsicht unproduktiv. Er schadet nicht nur dem Unternehmen selbst, sondern wirkt letztlich auch der Akzeptanz der Technologie entgegen. Ein neueres Beispiel dafür ist Java, das seine ursprüngliche Anziehungskraft verloren hat. Mit seinem Versuch, Java zu kontrollieren, hat Sun Microsystems es im Prinzip verloren. Java behauptet sich zwar immer noch recht gut, ist aber deutlich hinter seinem Potenzial zurückgeblieben.

Sun versuchte nicht, mit Java selbst Geld zu verdienen, aber das Unternehmen sah die Programmiersprache als Möglichkeit, die Individualität des Computers in den Augen der Anwender zu steigern und uns aus den Fängen von Microsoft zu befreien - und nebenbei auch noch mehr Sun-Hardware zu verkaufen. Aber obwohl die Firma der Versuchung widerstand, mit Java mehr Geld zu machen, glaubte es, die Sprache als Phänomen und Mikrobe kontrollieren zu müssen. Nur deswegen mussten alle Sun-Lizenzbedingungen eine Menge zusätzliches Gepäck mit sich herumschleppen.

Gutes Produkt. Aber leider versuchten die Sun-Leute zu sehr, Microsoft eins auszuwischen. Sie waren von Angst, Hass und Antipathie motiviert und legten ein Geschäftsgebaren an den Tag, das vielleicht Mitte bis Ende der 90er-Jahre mal angesagt war. (Nehmen Sie sich den

Song der Rockgruppe Grateful Dead zu Herzen: „Ain't no time to hate.") Und weil sie Microsoft so hassten und fürchteten, trafen sie lauter falsche Lizenzentscheidungen. Sie erschwerten es jedem, sogar ihren Partnern, das Produkt einzusetzen. **Deshalb beschlossen Unternehmen wie Hewlett Packard und IBM durch die Bank, eigene JavaImplementierungen zu entwickeln.** Ihre Devise lautete: „Zum Teufel mit Sun."

Sun versuchte, über zwei verschiedene Standardisierungsgremien eine Standardisierung von Java zu erwirken, und beide Male machten sie wegen Problemen bei Fragen der Kontrolle einen Rückzieher. Einerseits wollte Sun die Sprache standardisieren. Aber andererseits wollte es die Kontrolle nicht verlieren. Deshalb stellten die Standardisierungsgremien klar: „Hey, hier geht es nicht nur um euch." Daraufhin ließ Sun die Sache fallen. Das ist ein Beispiel für ein Unternehmen, das eine Technologie auf **eine Art und Weise** zu kontrollieren versuchte, die für die Anwender der Technologie keinen Sinn machte. Daran scheitert jedes Unternehmen. Und daran scheitert auch die Technologie selbst. Zumindest braucht sie länger, um akzeptiert zu werden.

Vergleichen Sie diese Herangehensweise mit der Wenn-du-etwas-liebst-gib-es-frei-Strategie, die Palm Computing einschlug. Die Leute bei Palm legten ihre Entwicklungsumgebung und ihre Plattform nicht nur Drittherstellern, sondern auch Einzelpersonen offen, die Programme für die Plattform schreiben wollen. Sie legten ihre APIs offen, und stellten ihre Entwicklungstools gratis zur Verfügung. **Auf diese Weise entstand rund um den Palm Pilot herum eine kleine feine Industrie. Mit einem Mal war das Palm-Phänomen nicht länger auf eine einzelne Firma begrenzt, die einen neuen Markt beackert. Heute gibt es Firmen, die Spiele verkaufen, die auf Palm Pilots laufen, und raffiniertere Kalenderprogramme als die von Palm selbst angebotenen.** Die Kunden können frei wählen, was sie möchten, und das kommt jedem zugute, vor allem Palm selbst, das durch die **Öffnung in den Genuss eines größeren Marktes kam.**

Handspring schlug für sein Gerät, den Visor, den gleichen Weg ein.

Der Visor ist ein Palm-Konkurrent, der das Palm-Betriebssystem nutzt. Handspring treibt seine Offenheit noch einen Schritt weiter und lässt Hardware-Plug-ins wie GPS-Empfänger oder die Aufrüstung zum Mobiltelefon zu. Wie Palm ist Handspring dabei, eine Szene von Partnerunternehmen aufzubauen, die die neue Plattform unterstützen.

Was Sun anbetrifft, so hätte es jedem ohne Wenn und Aber erlauben können, sein eigenes Java zu entwickeln, und darauf setzen können, die Sache selbst besser zu machen. So würde sich ein Unternehmen verhalten, dessen Blick nicht durch Gier oder Angst vor dem Wettbewerb getrübt ist. So würde sich ein Unternehmen verhalten, das an sich glaubt. **Und das für Hass keine Zeit hat.**

Die Karussellfahrt, die vor uns liegt

Gibt es etwas Nervtötenderes als Trend-Gurus? Diese selbstverliebten Schnösel, die zu wissen vorgeben, wo uns die wahnwitzige technologische Karussellfahrt noch hinführen wird? Vermutlich haben sie eine Existenzberechtigung. Sie bevölkern die Podiumsdiskussionen und Eröffnungsreden der immer gleichen Technologiekonferenzen, die sich auszubreiten scheinen wie hässliche, ungenießbare Pilze im Blumenbeet. In der Hoffnung, Technologietrends ausbeuten zu können, geben die Teilnehmer von Technologiekonferenzen für die Vorträge solcher Gurus Tausende von Dollars aus. Aber weil Trend-Gurus eine Armee von Hotelangestellten, Köchen und Barkeepern in Lohn und Brot halten, haben sie vermutlich eine Existenzberechtigung.

Und jetzt meint David, ich solle auch eines dieser Kapitel über die Zukunft unserer Wirtschaft verfassen. Mir graut vor dem Gedanken, aber, hey, er hat mich beim Boogie-Boarden nicht ertrinken lassen, und wenn er meint, die Leser würden lieber etwas über unsere wirtschaftliche Zukunft als über den Sinn des Lebens erfahren, halte ich die Klappe und tue, was er sagt.

Aber trotzdem.

Ich erkläre hiermit, dass ich, soweit ich mich erinnern kann, in meinem ganzen Leben noch keine besonders guten Voraussagen getroffen habe. Habe ich etwa vorausgesehen, dass mein kleines Betriebssystem, das ich zunächst nur für den Eigengebrauch geschrieben habe, eines Tages auf der ganzen Welt zu finden sein würde? Bestimmt nicht. Das hat mich selbst überrascht. Und wie.

Ich kann zu meiner Verteidigung nur sagen, dass auch sonst niemand diesen Kristallkugel-Hokuspokus sonderlich gut zu beherrschen scheint, und während ich nur überrascht war, als Linux zum Branchenliebling aufstieg, war jeder andere völlig platt. Deshalb bin ich der Sache alle Mal näher gekommen als die meisten. Und wer weiß? Viel-

leicht gehe ich mit diesem Kapitel als der Nostradamus unserer Zeit in die Geschichte ein.

Aber vielleicht auch nicht. Okay, jetzt aber los.

Wir können natürlich auf Erfahrungen der Vergangenheit zurückblicken und im traurigen Detail verfolgen, wie ein unbezwingbar erscheinendes Unternehmen wie, sagen wir, AT&T den Biss verlor. Aber das zeigt uns nur, dass das Unkraut, wenn wir nur lange genug am Markt sind, eines Tages auch diese hübschen grünen Gebäude in Redmond überwuchern wird. So wie das kippe junge Starlet von heute Falten und einen schlaffen Busen bekommen wird, wird der Vorzeigeunternehmer von heute durch ein neues inspirierteres Modell abgelöst werden, und sein Unternehmen wird, selbst wenn es alles daran setzt, sich neu zu erfinden - oder wie immer das jeweilige Modewort des Monats gerade lauten mag - genauso morsch und brüchig enden wie AT&T.

Das ist die Evolution. Man muss kein Genie sein, um das zu erkennen. Kein Unternehmen lebt ewig, und das ist gut so.

Aber wovon wird diese Evolution getrieben? Ist Technologie mit einer fundamentalen inhärenten Evolution verbunden, die eines Tages dazu führen wird, dass die Computer übernehmen und die menschliche Rasse im Staub versinkt, wie einige Leute zu glauben scheinen? Oder wird der technologische Fortschritt einfach nur durch eine Art unvermeidliche Vorwärtsbewegung verursacht - nach der Devise „Vorwärts marsch und schert euch nicht um die Torpedos.“

Ich sage nein. Technologie ist das, was wir daraus machen, und weder die Wirtschaft noch die Technik werden die grundlegenden Bedürfnisse und Sehnsüchte der Menschen verändern. Langsam aber sicher wird die Evolution genau wie in anderen Bereichen dazu führen, dass die Technologie immer weniger dem reinen Überleben und immer mehr der gesellschaftlichen Kommunikation und schließlich dem Vergnügen dient. (Dejà-Vu-Warnung: Ja, Sie sind dieser Theorie in diesem Buch bereits begegnet und werden, falls Sie bis zum bitteren Ende durchhalten, noch einmal darauf stoßen.)

Menschen sind von Natur aus Partylöwen, und die Technologie wird dem Rechnung tragen.

Vergessen Sie also alle Vorhersagen über die technischen Möglichkeiten in zehn Jahren. Sie sind nicht sonderlich relevant. Wir waren vor

dreißig Jahren imstande, Menschen auf den Mond zu schicken, und sind seither nicht wieder dorthin zurückgekehrt. Meiner Meinung nach liegt das einfach daran, dass der Mond sich als freudloser Ort

praktisch ohne jedes Nachtleben
diesem Grund *wollten* die Leute

erwies - so ähnlich wie San Jose.
Aus
nicht wieder dorthin, ungeachtet der
der Zwischenzeit zusammengerafft

ganzen Technologie, die wir i n

Was Sie bei Ihren Überlegungen zur Zukunft der Technologie wirklich berücksichtigen müssen, sind die Wünsche und Bedürfnisse der Menschen. Sobald Sie sich darüber im Klaren sind, bleibt nur eine Frage offen: Wie schnell können Sie das Ding in Massenproduktion herstellen und so billig anbieten, dass die Leute es sich leisten können, ohne auf etwas *anderes* verzichten zu müssen, das sie auch haben *wollen*. Nur darauf kommt es an.

Hier ist ein kurzes Abschweifen angebracht. Was sich wirklich verkauft, ist natürlich nicht die tatsächliche, sondern die *wahrgenommene* Wirklichkeit. Kreuzfahrtschiffe verkaufen die Wahrnehmung von Freiheit, Meer und Salzwasser, gutem Essen und TraumschiffRomantik. Wen kümmert die Enge in der Kabine, wenn er sich frei wie ein Vogel *fühlt*?

Was bedeuten diese Beobachtungen? Sie erklären zum Beispiel, warum die Leute völlig ausrasten wegen der Sony PlayStation 2, jener technologischen Meisterleistung, die in diesem Jahr in die Geschäfte kam. (Ich schreibe das nur wenige Tage nach der Einführung der PlayStation 2 in den USA Ende Oktober 2000.) Mehr brauche ich zum Thema Spaßgesellschaft wohl nicht zu sagen.

Sie zeigen darüber hinaus sehr deutlich, inwiefern PCs ein Wahrnehmungsproblem haben: Natürlich macht die Spielkonsole die PC-Branche nervös, vor allem weil Spielkonsolen als unbedrohlicher Spaß wahrgenommen werden, während PCs vor allem als kompliziert und teuer gelten. Und manchmal sogar als feindlich gesonnen.

Und schließlich bestärken sie mich in meiner persönlichen Überzeugung, dass etwas grundfalsch gelaufen sein muss, falls Betriebssysteme in fünfzehn Jahren immer noch so ein großes Thema sein sollten. Das mag aus dem Munde eines Menschen, der seinen Ruhm hauptsächlich daraus bezieht, dass er sich ein eigenes Betriebssystem geschrieben hat, seltsam klingen. Aber Tatsache ist, dass statistisch gesehen *niemand* ein Betriebssystem haben will.

Tatsächlich will auch niemand einen Computer haben. Was alle wollen, ist ein magisches Spielzeug, mit dem man im Internet surfen, Hausarbeiten schreiben, Spiele spielen, das Konto ausgleichen kann usw. Die meisten Leute würden die Tatsache, dass man dazu einen Computer und ein Betriebssystem braucht, am liebsten ignorieren.

Aus diesem Grund sind viele Analysten so begeistert von Geräten wie der Sony Playstation 2, die teilweise die Aufgaben eines Computers übernehmen, ohne ihren Anwendern wegen ihrer erschreckenden Kompliziertheit den Schweiß auf die Stirn zu treiben. Rein technisch betrachtet ist das zwar Unsinn, weil auf diese Weise immer noch mehr Computer in unser Haus Einzug halten werden, aber uns ist einfach *nicht bewusst*, wie abschreckend kompliziert auch sie sein können.

Deshalb wette ich darauf, dass Sony das nächste Microsoft sein wird, vorausgesetzt, es stellt die Weichen richtig. Ich will nicht behaupten, diese Vorhersage sei von Nostradamus-artiger Umwerfendheit (Ja, ich weiß: Dieses Wort gibt es wahrscheinlich nicht. Sollte es aber!), weil auch andere ihr zustimmen würden, aber ich werde versuchen, die *Gründe* dafür zu erklären.

Ich sage nicht das Ableben des PCs voraus, wie so viele vor mir es vergebens getan haben. Der PC hat nach wie vor seine grundlegenden Stärken: Er ist das Schweizer Taschenmesser unter den Computern. Erkennbar kompliziert genug, um Leute abzuschrecken, die Technik nicht mögen. Kompliziert, weil er eben nicht für eine einzige Sache maßgeschneidert ist. Genau diese Flexibilität macht ihn attraktiv.

Und dann die eine Sache, die alles steuert und insgeheim verbindet: Kommunikation. Überall. Sie können nicht leben, ohne mindestens zweimal pro Stunde nach Ihrer E-Mail zu schauen? Kein Problem, mein E-Mail-süchtiger Freund. Sie können sich mit leichtem Schuldgefühl einen freien Tag am Strand gönnen und sich doch ständig über die Ereignisse im Büro auf dem Laufenden halten. Sie wissen ja: Was sich verkauft, ist nicht die *Realität*, Urlaub zu haben, sondern das *wahrgenommene* Gefühl von Freiheit. Am Ende spielt die Größe dann doch eine Rolle, und sei es nur, um all die Wunderwerke der Technik trivial und unbedrohlich erscheinen zu lassen.

Und welche Rolle spielen Linux und Open Source bei alledem? Das werden Sie gar nicht merken. Linux wird in diesen Sony-Geräten stecken. Sie werden es nie zu sehen bekommen, Sie werden es nie kennen lernen, aber es wird da sein und alles am Laufen halten. Es wird in

dem Mobiltelefon stecken, das gleichzeitig als Ihr ganz persönlicher Kommunikationsknoten für den Rest Ihrer elektronischen Wichtel fungiert, wenn Sie sich nicht in der Nähe Ihres lokalen Funknetzes befinden.
Sie werden es erleben. Es ist nur eine Frage der Zeit. Und des Geldes.

Warum Open Source einen Sinn ergibt

IBM ist ein Unternehmen, das die Leute von jeher über den Tisch zog. Es verdiente sein Geld damit, ein unfreiwilliges Publikum an sich zu binden und sicherzustellen, dass niemand sonst einen Fuß in die Tür bekam. Tatsächlich arbeiteten die meisten Computerfirmen so. Manche von ihnen tun es immer noch. Als IBM dann den Personal Computer entwickelte, legte es ungewollt seine Technologie offen, so dass jeder sie reproduzieren konnte. Dieser eine Akt trug mehr als alles andere dazu bei, die PC-Revolution voranzutreiben, die ihrerseits die Informationsgesellschaft, die Internetrevolution, die New Economy vorantrieb - oder wie die ungeheuren Veränderungen, die zurzeit auf der ganzen Welt stattfinden, auch sonst immer heißen mögen.

Die Geschichte des PCs illustriert wie keine andere die grenzenlosen Vorteile, die sich aus der Open-Source-Philosophie ergeben. Auch wenn der PC nicht nach dem Open-Source-Modell entwickelt wurde, ist er ein Beispiel für eine Technologie, die jeder Einzelperson oder jedem Unternehmen zugänglich gemacht wurde, um sie zu klonen, zu verbessern und zu verkaufen. In ihrer reinsten Form gestatten OpenSource-Projekte jedem, sich an ihrer Entwicklung oder kommerziellen Verwertung zu beteiligen. Linux ist offensichtlich das erfolgreichste Beispiel dafür. Was in meinem chaotischen Jugendzimmer in Helsinki begann, wuchs zum größten Gemeinschaftsprojekt der Weltgeschichte heran. An seinem Anfang stand eine Ideologie, die von Softwareentwicklern geteilt wurde, die daran glaubten, dass Computerquellcode frei ausgetauscht werden sollte, wobei die General Public License - das Anti-Copyright - als mächtiges Werkzeug der Bewegung diente. Diese Ideologie entwickelte sich zu einer Methode für die kontinuierliche Entwicklung der jeweils besten Technologie. Diese wiederum gewann eine breite Marktakzeptanz, die sich zum Beispiel daran zeigte, dass Linux im Schneeballsystem als Betriebssystem für Webserver übernommen wurde, und an den unerwartet einträglichen Börsengängen.

Was zunächst ideologisch inspiriert war, hat seine technologische Stärke bewiesen und bewährt sich am Markt. Heute breitet sich der Open-Source-Gedanke auch jenseits von Technik und Wirtschaft aus. An der juristischen Fakultät der Harvard-Universität haben die Professoren Larry Lessig (der heute in Stanford lehrt) und Charles Nesson das Open-Source-Modell auf das Rechtswesen übertragen. Sie begannen das Open Law Project, das sich auf ehrenamtlich arbeitende Anwälte und Jurastudenten stützt, die Meinungen und Rechercheergebnisse auf der Website des Projekts veröffentlichen und so die Vorbereitung von Anträgen und Schriftsätzen unterstützen, die das amerikanische Gesetz zur Erweiterung des Urheberrechts (United States Copyright Extension Act) anfechten. Dahinter steht die Theorie, dass die stärksten Argumente entstehen, wenn möglichst viele juristische Köpfe an einem Projekt zusammenarbeiten und durch Postings und deren Beantwortung ein Berg von Informationen zusammenkommt. Die Website fasst die Vor- und Nachteile gegenüber der herkömmlichen Vorgehensweise prägnant zusammen: „Wir gehen davon aus, dass der Gewinn an Darstellungstiefe und Argumentationsbreite den Verlust an Geheimhaltung aufwiegen wird.“ (In einen anderen Kontext übertragen: „Millionen von Augen bringen alle Softwarefehler zum Verschwinden.“)

Das widerspricht zwar jahrelangen Gepflogenheiten der wissenschaftlichen Recherche, ist aber in vielerlei Hinsicht sinnvoll. Sie brauchen sich nur vorzustellen, wie dieser Ansatz die Entwicklung medizinischer Heilverfahren beschleunigen könnte. Oder welche Stärke die internationale Diplomatie entwickeln könnte, wenn die besten Köpfe hinter ihr stünden. Je kleiner die Welt wird, je mehr sich Lebenstempo und Wirtschaft beschleunigen und je mehr Technologien und Informationen zur Verfügung stehen, desto klarer zeichnet sich ab, dass die Methode der zugehaltenen Taschen immer weniger zeitgemäß ist.

Hinter Open Source steht eine sehr einfache Theorie. Im Fall eines Betriebssystems ist der Quellcode - die dem System zugrunde liegenden Programmieranweisungen - frei zugänglich. Jeder kann ihn verbessern, verändern, verwerten. Aber diese Verbesserungen, Veränderungen und Verwertungen müssen ihrerseits frei zugänglich gemacht werden. Denken Sie an Zen. Das Projekt gehört keinem und jedem. Indem es allen offen steht, sind schnelle, ständige Verbesserungen möglich. Wo Teilnehmerteams parallel arbeiten, lassen sich viel

schneller und erfolgreicher Ergebnisse erzielen, als hinter verschlossenen Türen.

Genau das haben wir bei Linux erlebt. Stellen Sie sich vor: Anstelle eines winzigen, abgeschotteten Entwicklungsteams, das seine Arbeit geheim hält, steht Ihnen ein Monster zur Seite. Potenziell tragen Millionen der hellsten Köpfe zu einem Projekt bei und werden dabei durch einen Überprüfungsprozess durch Gleichgesinnte unterstützt, der, ähm, seinesgleichen sucht.

Wenn Leute zum ersten Mal von Open Source hören, klingt der Gedanke grotesk. Deshalb hat es auch jahrelang gedauert, bis die Botschaft von seinen Vorteilen Gehör fand. Das Modell hat sich nicht wegen seiner Ideologie verkauft. Es erregte erst Aufmerksamkeit, als sich zu zeigen begann, dass Open Source die beste Methode zur Entwicklung und Verbesserung der Technologie mit der höchsten Qualität ist. Und jetzt ist diese Technologie gerade dabei, den Markt zu gewinnen. Erst diese jüngste Leistung hat wirklich zu einer Akzeptanz des Open-Source-Gedankens geführt. Firmen konnten zahlreiche Zusatzdienstleistungen rund um Linux herum anbieten oder Open Source als Möglichkeit nutzen, eine Technologie populär zu machen. Wenn der Rubel rollt, sind die Leute leicht zu überzeugen.

Zu den rätselhaftesten Teilen des Open-Source-Puzzles gehört für die meisten Leute die Frage, was so viele gute Programmierer dazu bringt, ohne einen Pfennig Geld zu arbeiten. Das Zauberwort heißt Motivation. In einer Gesellschaft, in der das Überleben mehr oder weniger sichergestellt ist, ist Geld nicht der Hauptmotivator. Es ist nachgewiesen, dass Menschen am meisten leisten, wenn sie von einer Leidenschaft getrieben sind. Wenn sie Spaß an der Sache haben. Das gilt für Dramatiker, Bildhauer und Unternehmer ebenso wie für SoftwareIngenieure. Das Open-Source-Modell gibt den Leuten die Chance, ihre Leidenschaft auszuleben. Und dabei Spaß zu haben. Und mit den besten Programmierern der Welt zusammenzuarbeiten, nicht den wenigen, die zufällig in der gleichen Firma wie sie beschäftigt sind. Open-Source-Entwickler streben danach, sich die Achtung ihrer Peers zu erwerben. Das muss hoch motivierend sein.

Anscheinend kann Bill Gates das nicht verstehen. Oder ist ihm die fiese rhetorische Frage, die er einst stellte, heute womöglich doch peinlich? „Unter anderem verhindern Sie, dass gute Software geschrieben wird. Wer kann es sich leisten, ohne Gegenleistung eine professionelle

Arbeit zu erbringen?", schrieb er in einem Brief an Open-Source-Programmierer.

Übrigens lässt sich das Open-Source-Phänomen auch verstehen, wenn man sich vor Augen führt, wie die Religion vor vielen Jahrhunderten (einige Zeitgenossen tun es heute noch) auf die Wissenschaft reagierte. Die Wissenschaft wurde anfangs als etwas Gefährliches, Subversives, gegen das Establishment gerichtetes gesehen - im Grunde so, wie Softwarefirmen manchmal Open Source betrachten. Aber genau so wenig wie die Wissenschaft aus dem Versuch heraus entstand, die etablierten Kirchenführer zu unterminieren, wurde das Open-Source-Modell ersonnen, um das Software-Establishment in die Luft zu sprengen. Es geht darum, die bestmögliche Technologie hervorzubringen, das ist alles.

Mit Wissenschaft allein lässt sich kein Geld verdienen. Es sind die Sekundärwirkungen der Wissenschaft, die all den Wohlstand hervorbringen. „Das Gleiche gilt für Open Source. Es lässt Sekundärindustrien entstehen, die die etablierten Unternehmen herausfordern, auf ähnliche Weise, wie die Ergebnisse der Wissenschaft die Kirche herausforderten. Kleine Firmen wie VA Linux sind dank Open Source plötzlich in der Lage, mit traditionellen Firmen zu konkurrieren. Wie Sir Isaac Newton schon sagte: Sie stehen auf den Schultern von Giganten.

Und ja, je mehr Open Source in der Weltwirtschaft an Boden gewinnt und je mehr Anerkennung seine Entwickler genießen, desto mehr steigt ihr Marktwert als Mitarbeiter. Unternehmen durchforsten die Credit-Lists, die die Beiträge zu Open-Source-Software von jeher begleiten, um festzustellen, wer mehrere Beiträge vorzuweisen hat. Und dann beauftragen sie ihre Personalabteilungen, schubkarrenweise Geld und Aktienoptionen aufzufahren. Ich habe weiter oben verkündet, Geld sei nicht der Hauptmotivator und, nein, ich habe meine Meinung in den darauf folgenden Sätzen nicht geändert. Aber ich muss sagen, es ist nicht schlecht, für harte Arbeit mit Geld belohnt zu werden. Auf jeden Fall kann ich es gut gebrauchen, wenn ich mit meinem BMW zum Tanken fahre.

Wie die Wissenschaft selbst sind auch die Nebenwirkungen des Open-Source-Modells endlos. Es schafft Dinge, die bis vor kurzem als unmöglich galten und eröffnet unerwartete neue Märkte. Mit Linux und anderen Open-Source-Projekten können Unternehmen ihre eigenen Versionen entwickeln und eigene Veränderungen vornehmen - und

das ist auf einem anderen Weg einfach nicht möglich. Es ist interessant zu sehen, dass so gut wie nichts von dem, was je mit Linux gemacht wurde, im Vorfeld absehbar war. Zum Beispiel, dass Linux in China Fuß fassen würde. Bisher hat sich die Softwareentwicklung in Asien hauptsächlich auf die Übersetzung amerikanischer oder europäischer Software beschränkt. Jetzt nutzen Menschen in diesem Teil der Welt Linux, um ihre eigene Software zu entwickeln. Und ich erinnere mich mit Stolz an diesen Typ, der mich auf der Comdex ansprach, um mir die Zapfsäule zu zeigen, auf der Linux lief. Dann führte er mir den Prototyp einer mit Linux ausgestatteten Zapfsäule vor, auf der er Webbrowser installieren wollte, damit die Kunden in den drei Minuten, in denen ihr Tank gefüllt wurde, vielleicht bei CNN.com vorbeischaun konnten. Auf den Schultern von Giganten stehen.

Es ist inspirierend, dass Leute Technologien wie Linux einsetzen, nur um eine bessere Zapfsäule zu entwickeln. Diese Art von Innovation wäre in den engen Grenzen eines Unternehmens sehr wahrscheinlich nicht möglich gewesen, denn wenn du als Unternehmen Linux auf den Markt brächtest, würdest du den offensichtlichen Weg einschlagen, und der ist momentan der Servermarkt oder der Markt für HighendDesktops. Dagegen erlaubt Open-Source-Software den Unternehmen, selbst zu entscheiden, was sie tun wollen. Deshalb wird Linux auch in Embedded Devices eingesetzt. Das TiVo-System läuft mit Linux, der Transmeta Web Slate läuft mit Linux, Telephony läuft mit Linux. So werden mit Open Source Milliarden von Dollars verdient.

Es ist, als würde man dem Universum erlauben, selbst für sich zu sorgen. Indem du die Technologie nicht kontrollierst, schränkst du ihre Nutzung nicht ein. Du stellst sie zur Verfügung, die Leute treffen Entscheidungen vor Ort und nutzen die Technik als Sprungbrett für eigenen Produkte und Dienstleistungen. Und weil die meisten dieser Entscheidungen keinen höheren Sinn ergeben müssen, funktionieren sie wirklich gut. Es kommt nicht darauf an, die Verbreitung von Linux zu fördern. Es kommt darauf an, Linux zugänglich zu machen und darauf zu warten, dass es sich von ganz alleine verbreitet. Das gilt übrigens nicht nur für Linux, sondern für jedes offene Projekt.

Open Source ergibt einen Sinn.

Niemand bestreitet die Notwendigkeit der freien Rede. Menschen haben diese Form der Freiheit mit ihrem Leben verteidigt. Freiheit ist immer etwas, was du mit deinem Leben verteidigen musst. Aber diese

Entscheidung ist anfangs nicht einfach zu treffen. Mit Offenheit ist es genau so. Du musst dich zu der Entscheidung durchringen, dass du offen bist. **Diesen Standpunkt einzunehmen, ist anfangs schwierig, aber am Ende gewinnst du mehr, nicht weniger Stabilität.**

Denken Sie an die Politik. Würde man die gleiche Logik, mit der gegen **Open Source** argumentiert wird, auf **Regierungen anwenden, hätten wir immer Einparteiensysteme.** Offensichtlich ist ein Einparteiensystem sehr viel einfacher zu handhaben als ein Mehrparteiensystem, das offene politische System, nach dem ein großer Teil der Welt arbeitet. Wenn du nur eine Partei hast, entfällt der Stress, dich mit anderen Leuten abzustimmen. Daraus könnte man ableiten, die Regierungsarbeit sei zu **wichtig, um sie durch das Geben und Nehmen zu belasten, das in einem offenen System erforderlich ist. Aus irgendeinem Grund sehen die Leute den Trugschluss dieses Arguments, wenn es um Politik und Regierungen geht, aber nicht, wenn es um Wirtschaft geht. In der Wirtschaft macht Offenheit die Leute ironischerweise nervös.**

Die Argumente, mit denen Unternehmen eine Öffnung ihrer Technologie verhindern, sind überzeugend. Das haben wir noch nie so gemacht, sagt das Management. Und das ist erschreckend. **Die Leute haben vor Veränderungen Angst, unter anderem, weil ihr Ausgang ungewiss ist. Indem sich ein Unternehmen an den Status quo klammert, kann es seine Entwicklung besser abschätzen, und manchmal ist ihm das wichtiger als enorm erfolgreich zu sein. Unternehmen, die so denken, werden vorhersehbar erfolgreich sein, statt unvorhersehbar wirklich, wirklich, wirklich erfolgreich zu sein.**

Es ist für ein Unternehmen nicht einfach, ein vorhandenes Produkt in ein **Open-Source-Produkt zu verwandeln.** Es gilt eine Fülle heikler Fragen zu bedenken. **Zum einen hat das Unternehmen im Lauf der Monate oder Jahre, in denen es das Produkt entwickelt hat, eine Menge internes Wissen aufgebaut. Dieses geistige Eigentum ist das Pfund, mit dem das Unternehmen wuchert. Die Organisation ist nicht bereit, das geistige Eigentum preiszugeben, das sie am Leben erhält. Gleichzeitig bedeutet dieses interne Wissen auch eine Barriere für Außenseiter. Es entmutigt sie, sich einzubringen.**

Ich kenne jedoch Unternehmen, die den Schritt von der Geschlossenheit in die Offenheit geschafft haben. Zu ihnen gehört Wapit, ein finnisches Unternehmen, das den Service und die Support-Infrastruktur für verschiedene interaktive Geräte bereitstellt. Bei dem fraglichen

Projekt ging es um den Webserver des Unternehmens, der wie eine Art Wandtelefon funktioniert. Für Wapit ist die Entscheidung, seine Software frei zugänglich zu machen, absolut sinnvoll. Sie wollen ihren Service aufbauen, müssen dazu aber zuerst ihre Infrastruktur erweitern. Zu diesem Zweck muss eine Menge Software geschrieben werden. Das ist ein notwendiges Übel. Deshalb betrachteten sie ihre Entscheidung nicht als Preisgabe ihres geistigen Eigentums an andere, sondern sehen es so: Es kostet viel Entwicklerzeit, die benötigte Software zu schreiben, ohne dass es zur Wertschöpfung des Unternehmens beitragen würde, sich eifersüchtig daran zu klammern.

Mehrere Umstände kamen Wapit zugute: Erstens ging es nicht um ein riesiges Projekt. Zweitens wurde die Entscheidung, die Software als **Open-Source-Software zu lizenzieren, in der Anfangsphase nach der Unternehmensgründung getroffen.** Das Management führte sich vor Augen, zwar über die internen Ressourcen zur Entwicklung des Produkts zu verfügen, eigentlich aber eine größere Lösung anzustreben, als intern entwickelt werden konnte. Darüber hinaus sah es in der Verwendung eines **Open-Source-Ansatzes eine gute Möglichkeit, Wap als Standard zu fördern, den andere nutzen können. Das Unternehmen holte sehr frühzeitig meinen Rat ein, und ich sagte ihnen, sie müssten die Neigung bekämpfen, Entscheidungen intern zu treffen. Ich schlug vor, Außenstehende von wichtigen Besprechungen nicht fern zu halten. Wenn der Entscheidungsprozess weiterhin als Firmenangelegenheit behandelt werden würde, bestünde die Gefahr, Außenstehende vor den Kopf zu stoßen, denen es schwer fallen dürfte, sich gegen das Old-Boys-Network des Unternehmens zu behaupten. Das ist übrigens eines der Hauptprobleme, wenn Unternehmen Open-Source-Projekte aufbauen und pflegen. Es ist einfach, Lippenbekenntnisse für Open Source abzulegen. Tatsächlich aber kann sehr leicht eine Zwei-Klassen-Gesellschaft entstehen: Wir gegen sie. Viele Entscheidungen werden informell getroffen - wenn in der Cafeteria die Optionen diskutiert werden und ein Konsens entsteht, ohne dass die Diskussion je nach außen getragen wurde. Leute von außen mit stichhaltigen Meinungen werden im Prinzip durch die Tatsache niedergestimmt, dass die Entscheidung bereits in der Cafeteria des Unternehmens vorweg genommen wurde.**

Unter diesem Problem litt auch Neucap in den Monaten nach seiner beifälligen aufgenommen Entscheidung des Frühjahrs 1998, den Quell-

code (namens Mozilla) seiner nächsten Browsergeneration offen zu legen. Es dauerte lange, bis das Unternehmen seinem Open-Source-Versprechen wirklich gerecht wurde. Erst einmal mutierte es zu einem Lager von Neucap-Insidern, die kleine Patches von Außenstehenden nicht akzeptierten. Intern kannte jeder jeden. Und wenn sie nicht leibhaftig im Coffee-Shop saßen, um sich abzusprechen, dann bildeten die Insider im virtuellen Coffee-Shop eine eng verschworene Gemeinschaft. Diese Entwicklung führte zu einer ziemlich schlechten Presse und verhinderte, dass Mozilla als erste große Anstrengung wahrgenommen wurde, ein bestehendes kommerzielles Projekt frei zugänglich zu machen. Als sich seine Inaktivität herumsprach, konnte Netscape sich moralisch nicht länger aufs hohe Ross setzen. Das half dem Unternehmen, sich Außenseitern zu öffnen. Inzwischen scheint das Projekt sehr viel dynamischer zu verlaufen.

Wenn Leute erstmals von der Möglichkeit hören, ein bestehendes kommerzielles Projekt zu öffnen, stellen sie in der Regel die immer gleichen Fragen. Eine davon lautet, wie die Mitarbeiter im Unternehmen es empfinden, wenn ein Außenstehender eine bessere Arbeit abliefert als sie - und zwar auf so öffentlich sichtbare Weise. Ich denke, sie sollten das fabelhaft finden - fabelhaft, dass sie bezahlt werden, ohne den Großteil der Arbeit zu leisten. In dieser Hinsicht kennt Open Source, genau wie alles andere, was offen ist, keine Gnade. Es zeigt, wer seinen Job beherrscht, wer besser ist. Du kannst dich nicht hinter Managern verstecken.

Open Source ist die beste Möglichkeit, zusätzliches Talent für das Unternehmen zu gewinnen. Trotzdem braucht man jemanden im Unternehmen, der die Bedürfnisse des Unternehmens im Auge behält. Diese Person muss nicht einmal der Projektleiter sein. Im Prinzip könnte es für das Unternehmen sogar von Vorteil sein, wenn ein Außenstehender die Aufgabe übernimmt - zum Nulltarif. Solange er den besseren Job macht, spricht nichts dagegen. Allerdings besteht die Gefahr, dass der Außenstehende das Projekt in eine Richtung lenkt, die den Bedürfnissen des Unternehmens entgegensteht. Deshalb muss das Unternehmen darauf achten, seine Interessen zu wahren. Die Öffnung des Projekts mag der Organisation die Möglichkeit geben, ihre Ressourcen zurückzufahren. Trotzdem kann sie ihre Ressourcen vor Ort nicht völlig abbauen. Das Open-Source-Projekt nimmt möglicherweise eine sehr viel größere Dimension an als ein Projekt, das rein

intern durchgeführt wird. Ressourcen von außen lassen es zu einem billigeren, vollständigeren und ausgewogeneren System werden. Aber es gibt da eine Kehrseite: Es berücksichtigt nicht mehr ausschließlich die Interessen des *Unternehmens*. Möglicherweise berücksichtigt es sogar die Interessen der Kunden.

Das Schwerste an alledem ist wahrscheinlich, die Kontrolle abzugeben und die Tatsache zu akzeptieren, dass Außenstehende es womöglich besser können.

Die andere Schwierigkeit besteht darin, einen starken technischen Leiter im Unternehmen zu finden. Er muss ein Mensch sein, dem die anderen auf zwei Ebenen vertrauen - technisch und politisch. Er muss jemand sein, der zugeben kann, dass das Projekt möglicherweise von Anfang an falsch angelegt war. Statt solche Probleme zu verdrängen, muss er in der Lage sein, alle Beteiligten davon zu überzeugen, dass es am besten ist, noch einmal von vorn anzufangen und das bisher Geleistete zu vergessen. Das ist nicht die Art von Botschaft, die die Leute gerne hören. Aber wenn sie von jemandem kommt, den sie respektieren, werden sie sie akzeptieren.

Angesichts der zu erwartenden politischen Rangeleien und der typischen Unternehmensabläufe muss der technische Leiter eine ziemlich starke Persönlichkeit besitzen. Er sollte jemand sein, der gerne per E-Mail arbeitet und es vermeidet, Partei zu ergreifen. Ich möchte ihn nicht als „Verbindungsoffizier“ bezeichnen, weil das zwei feindliche Lager implizieren würde - die interne Seite und die externe Seite. Und so sollte es nicht sein. Dieser technische Leiter wird vom Unternehmen dafür bezahlt, Open Source zu machen. Ihm selbst und allen anderen Beteiligten ist klar, dass er nicht dafür bezahlt wird, seinen Kollegen in der Firma nach dem Mund zu reden. Er wird dafür bezahlt, das Projekt abzuwickeln. Es besteht eine gewisse Gefahr, dass der Leiter zu eng mit der Firma verbunden ist. Möglicherweise vertrauen die Leute seinen technischen Fähigkeiten, aber nicht seinem nicht-technischen Urteilsvermögen.

Was gebraucht wird, ist ein Diplomat.

Das ist, als würde man sagen: „Findet mir einen ehrlichen Menschen.“

Aus diesem Grund habe ich im Lauf der Jahre alles daran gesetzt, nicht in Linux-Firmen hineingezogen zu werden. Jetzt, wo Geld zu fließen beginnt, ist das von immer größerer Bedeutung. Wenn so viele

Dollar im Spiel sind, fangen die Leute an, deine Motive zu hinterfragen. Für mich ist es hilfreich, dass ich von jeher als neutral galt. Sie können sich nicht vorstellen, wie wichtig es mir ist, diese Neutralität auch künftig zu wahren. Das Bemühen darum treibt mich zum Wahnsinn.

Okay. Sie haben Recht. Ich sollte mit der Predigt aufhören. Open Source ist nicht für jeden oder jedes Projekt oder jedes Unternehmen geeignet. Aber je mehr die Leute sich über den Erfolg von Linux klar werden, desto mehr erkennen sie, dass hier nicht ein paar idealistische, ungewaschene High-School-Kids große Töne spucken. Man braucht sich nur zu öffnen, und die Chancen ergeben sich von selbst. Ich rede über Open Source, seit Journalisten mich danach fragen - das heißt, seit ungefähr fünf Jahren. Früher musste ich endlos erklären, was das Besondere daran sein soll. Und, ehrlich gesagt, manchmal war es eine Ochsentour. Als warte man durch den Schlamm.

Mittlerweile haben es die Leute begriffen.

Ruhm und Reichtum

„Findest du nicht, dass der Ruhm auch eine Last sein kann?“, fragen mich manche Leute. Sie dürfen mir glauben: diese „Last“ ist keine Last. Es macht Spaß, berühmt zu sein, und berühmte Leute, die etwas anderes behaupten, versuchen nur, nett zu sein und nicht-berühmten Leuten das Gefühl zu geben, es besser zu haben. Es gehört sich, dass du deinen Ruhm herunterspielst und darüber klagst, wie er dein Leben zerstört.

Aber geben wir es zu: Jeder träumt von Ruhm und Reichtum. Ich weiß, dass ich es tat. Als Teenager wollte ich ein berühmter Wissenschaftler werden. Wie Albert Einstein, aber besser. Wer wollte das nicht? Und wenn schon nicht Wissenschaftler, dann Formel-1-Fahrer. Oder Rockstar. Oder Mutter Teresa. Oder Präsident der Vereinigten Staaten.

Und es tat überhaupt nicht weh, dahin zu gelangen, wo ich jetzt bin. Zugegeben, ich mag nicht Albert Einstein sein, aber es ist ein gutes Gefühl für mich, etwas bewirkt zu haben und etwas Sinnvolles zu tun. Und dass ich dafür Anerkennung bekomme, macht die Sache noch besser. Wenn Sie also das nächste Mal jemanden treffen, der über seinen Ruhm und Reichtum jammert, hören Sie nicht hin. Er tut es nur, weil es sich so gehört.

Dann ist also alles gut? Natürlich nicht. Es hat ein paar Nachteile, bekannt zu sein. Ich werde zwar nicht auf der Straße erkannt (jedenfalls nicht sehr oft), aber unter den unzähligen E-Mails, die ich bekomme, findet sich gelegentlich auch eine, die wirklich schwer zu beantworten ist und die man auch schlecht ignorieren kann. Was sagst du jemandem, der dich bittet, die Grabrede für seinen Vater zu halten, dem du nie begegnet bist? Ich habe diese Mail nie beantwortet, und ich habe deswegen immer noch Schuldgefühle. Ihm war die Sache sehr wichtig, und mir einfach nur lästig.

Oder wie sagst du jemandem, der dich bittet, den Eröffnungsvortrag bei einer Konferenz zu halten, dass du wirklich weder Zeit noch Lust dazu hast? Wie vermittelst du den Leuten, dass du längst keine Antworten auf dem Anrufbeantworter mehr abhörst, ohne als rücksichtsloser Bastard zu wirken? Obwohl du letztendlich einer bist. Es ist nicht, als würde mir jedes Thema am Herzen liegen, nur weil mir das Thema, für das ich bekannt bin, am Herzen liegt: Linux.

Natürlich macht es dir irgendwann nichts mehr aus, einfach nein zu sagen. Oder Anfragen ganz zu ignorieren: Ich finde E-Mails unter anderem deshalb so genial, weil sie sich so bequem und mühelos ignorieren lassen - was ist schon eine E-Mail angesichts der paar hundert, die ich jeden Tag bekomme? Das Medium ist so unpersönlich, dass du nur selten ein schlechtes Gewissen hast, wenn du es einfach ignorierst. Es kommt vor (siehe oben), aber nicht sehr oft. Und selbst wenn du es nicht ignorierst, ist es viel einfacher, per Mail „nein“ zu sagen, als in einem Gespräch oder am Telefon.

Das Problem sind im Grunde die Erwartungen, die an bekannte Leute herangetragen werden. Und die Tatsache, dass es offensichtlich nicht möglich ist, all den Erwartungen gerecht zu werden - und dass du gleichzeitig das Gefühl hast, du solltest wenigstens versuchen, ihnen gerecht zu werden. Aus diesem Grund war es auch so nervenaufreibend für mich, dieses Buch zu schreiben - zu versuchen, ein einigermaßen persönliches Buch herauszubringen und gleichzeitig nicht die Leute zu enttäuschen, die etwas anderes erwarten.

Und einige der Erwartungen sind einfach nur dumm. Ich habe oft das Gefühl, manche Leute sähen mich am liebsten als eine Art Bettelmönch unserer Tage, der in Einsamkeit ein asketisches Leben fristet. Und das nur, weil ich es für eine gute Idee hielt, Linux im Internet frei und umsonst zugänglich zu machen, statt meine Software in der üblichen Weise kommerziell zu vertreiben. So kommt es, dass mich das Eingeständnis, dass ich gern Geld ausbebe und endlich meinen alten Pontiac Grand Am* gegen ein Auto austausche, das mehr Spaß macht,

Es gibt an einem Pontiac Grand Am nichts auszusetzen, und er ist ein gutes Auto. Aber er ist wahrscheinlich auch das durchschnittlichste Auto in den ganzen Vereinigten Staaten, und einige Journalisten scheinen es interessant zu finden, dass ich einen so peinlich normalen Wagen fahre. Mein Gott, er ist ja noch nicht einmal ein Japaner! Die Leute werden jeden Respekt vor

befangen und defensiv werden lässt. Damit kommen wir zur zweiten Frage, die mir für gewöhnlich nach der Frage zur „Last des Ruhms“ gestellt wird. Nämlich: „Wird der Erfolg Linus zu Kopf steigen und/oder Linux verderben?“ Werde ich mich in einen egoistischen, verzo-genen Gecken verwandeln, der Bücher über sich schreibt, nur weil er sich gedruckt sehen will und Geld braucht, um sein neues nutzloses Auto zu finanzieren.

Natürlich lautet die Antwort darauf: ja.

Man nehme einen Menschen, dessen Lebensphilosophie darin besteht, Spaß zu haben und etwas Interessantes zu tun, füge etwas Geld und Ruhm hinzu - und was meinen Sie, was dabei herauskommt? Ein Instant-Philantrop? Ich glaube, nein. Es war mir nie auch nur in den Sinn gekommen, Geld für einen guten Zweck zu spenden, bevor David während der Arbeit an diesem Buch mich danach fragte. Ich sah ihn verständnislos an. „Rettet die Wale“ war das Erste, was mir in den Sinn kam. Ich bin offensichtlich nicht dazu geboren, eine große finanzielle Verantwortung zu übernehmen.

Ändert der Erfolg etwas an der Sicht der Dinge? Ja, definitiv. Linux war ein völlig anderes Geschöpf, als nur fünfzig ausgesprochene Technikfreaks damit arbeiteten und nicht, wie heute, 25 Millionen Leute (oder wie hoch die aktuelle Zahl gerade sein mag), die es zumindest gelegentlich einsetzen. Und Linux war damals, als die Leute einzig und allein aus Spaß und Interesse daran mitwirkten, etwas ganz anderes -vollkommen frei von den heute so offensichtlich vorhandenen kommerziellen Interessen.

Das Gleiche gilt auch für Linus als Person. Die Dinge verändern sich, und wenn ich das Gegenteil behaupten würde, so würde das an den Tatsachen nichts ändern. Linux ist nicht mehr die gleiche Bewegung wie vor fünf Jahren, und Linus ist nicht mehr der gleiche Mensch wie damals. Zum Teil war die Arbeit an Linux ja gerade deshalb so interessant für mich, weil sie nicht gleich blieb und ständig neue Probleme aufwarf. Dabei ging es übrigens nicht nur um technologische Lösun-

mir verlieren, wenn ich zugebe, dass ich Stunden über der Auswahl der richtigen Farbe für mein neues Auto, einen viel weniger praktischen BMW Z3, gebrütet habe. Schließlich lautet meine Devise „Just for Fun“. Dieses Auto ist absolut und total nutzlos. Ich fahre es ausschließlich zum Spaß. Und genau so will ich es haben.

gen, sondern auch um die Veränderung der Bedeutung von Linux angesichts seines Erfolges. Das Leben wäre langweilig, wenn es nicht so wäre.

Ich würde mich deshalb nicht als „verwöhnt“ bezeichnen, sondern einfach sagen, dass sowohl Linux als auch ich selbst mit dem kommerziellen Erfolg „anders“ geworden sind. Ich würde nicht unbedingt sagen „erwachsener“ - ich denke, drei Kinder zu bekommen, war in dieser Hinsicht einschneidender. Nein, eben anders. Und in vieler Hinsicht besser. Aber auch weniger unschuldig. Linux war ursprünglich nur für Technikfreaks gedacht und ein sicherer Hafen für Geeks. Eine Bastion der Reinheit, in der die Technologie zählte und sonst nicht viel.

Heute trifft das nicht mehr zu. Linux ist immer noch sehr technisch geprägt, aber die Tatsache, dass wir Millionen von Anwendern haben, macht uns allen sehr bewusst, dass wir unsere Entscheidungen sehr viel sorgfältiger abwägen müssen. Abwärtskompatibilität ist plötzlich ein Faktor - und eines Tages, in zwanzig Jahren vielleicht, wird jemand des Weges kommen, sagen, genug ist genug, und ein eigenes Betriebssystem schreiben, das „Fredix“ heißt und nicht den ganzen Ballast seiner Geschichte mit sich herumschleppt. Und genau so sollte es sein.

Aber ich bin unglaublich stolz darauf, dass „Fredix“ - wenn es denn eines Tages des Weges kommen sollte - ein anderes Umfeld vorfinden wird. Was immer man auch sonst davon halten mag, Linux hat es geschafft, den Leuten die Augen für neue Formen der Zusammenarbeit zu öffnen. Es hat bewiesen, dass Open Source eine Möglichkeit darstellt, auf der Arbeit anderer aufzusetzen. Den Open-SourceGedanken gibt es seit langem, aber Linux ist es gelungen, ihn ins allgemeine Bewusstsein zu heben. Deshalb braucht Fredix, wenn es denn des Weges kommt, nicht mehr bei Null anzufangen.

Und so ist die Welt ein etwas besserer Ort geworden.

Oder natürlich auch „Diannix“. Anders als heute wird die Informatik in zwanzig Jahren hoffentlich keine reine Männerdomäne mehr sein ...

Fast ein Jahr, nachdem wir mit der Arbeit an diesem Buch begonnen hatten, besuchten Linus und ich an einem Freitagabend den Freizeitpark, wo man Autorennen fahren und Baseball-Bälle schlagen kann und wo wir unsere Kräfte schon vor Monaten gegeneinander gemessen hatten. Dieses Mal machte Linus mich in beiden Disziplinen fertig: Er fuhr schneller und traf besser. Als wir später zusammen in einem türkischen Restaurant aßen, führte ich meine lausige Leistung auf einen besonders frustrierenden Tag im Büro zurück. Er sah hoch und sagte:

„Tja, drei Monate musst du schon noch durchhalten.“

„Wieso das denn?“

„Kannst du dann nicht den ersten Anteil deiner Aktienoptionen verkaufen?“

Ich erwähne diesen Vorfall, weil Linus mir bei unserem früheren Ausflug in den gleichen Freizeitpark gestanden hatte, er müsse wegen seines schlechten Gedächtnisses Tove regelmäßig nach seinen eigenen Telefonnummern fragen. Und nun erinnert er sich plötzlich an den Aktienoptionsplan eines anderen, und weiß genau, wo er war, als ich ihm das erste Mal davon erzählte. Vor einem Jahr schien er sich in der Rolle des zerstreuten Professors zu gefallen, an dem Details über Unbedeutenderes als die Superstring-Theorie oder den Speicherplatz seines ersten Computers vorbeilaufen. Jetzt ist er unglaublich präsent.

Im Januar saßen wir in meinem alten Hot-Tub, und ich lästerte, der Kulturausschuss von Marin County hätte ihn gern als Spende für das historische Museum. Im August fragte er mich beiläufig: „Hey, wann überlässt du ihnen endlich diesen Hot-Tub?“ Er muss kein elektronisches Gerät konsultieren, um zu wissen, von wann bis wann Avuton

zu Besuch kommen wird. Anders als vor einem Jahr weiß er über persönliche Details von Freunden und Kollegen bestens Bescheid. Er weiß sogar, was bei meinen Freunden und Kollegen gerade so läuft. Und für einen Typen, der auf meine erste Frage zu dem Thema ant-

wortete: „Eigentlich erinnere ich mich an meine Kindheit nicht sehr gut“, entsinnt er sich nun erstaunlich genau: „Habe ich dir schon erzählt, wie peinlich ich es fand, als meine Mutter wollte, dass ich meinen Großvater um die 100 FM bitten sollte, die mir noch für meine erste Uhr fehlten?“

Diese neu gefundene Wachheit war aber nur ein Zeichen für die Veränderung, die in Linus in einem wichtigen Jahr seines Leben vorgegangen zu sein schien. Es waren Kleinigkeiten. Im November unternahmen wir den Familienausflug nach Los Angeles, der den Hintergrund für das Vorwort über den Sinn des Lebens bildet, unter anderem auch deshalb, weil die Torvaldsens in das Haus des finnischen Generalkonsuls in Brentwood eingeladen waren. Vor dem Trip hatte Linus glasige Augen, als er die Weinregale in einem SafewaySupermarkt in Santa Clara sichtete. „Hilf mir, einen Wein als Geschenk auszusuchen“, sagte er. „Ich verstehe nichts vom Wein.“ Zehn Monate später weiß er, welcher von zwei ähnlichen Cabernets in der Minibar des Bodega Bay Lodge besser den Action-Film begleitet, den wir uns in unserem Hotelzimmer ansehen wollen. Ich erwische ihn dabei, wie er den Wein kennerisch schwenkt, ehe er einen Schluck nimmt.

Und dann ist da noch die Sache mit dem Sport. Bei meinem ersten Besuch in Linus' Haus brachte er seinem Körper und seinem physischem Wohlbefinden die Gleichgültigkeit des typischen Geeks entgegen: „Mein Körper ist nur dazu da, meinen brillanten Kopf zu tragen“, schien seine Philosophie zu sein. Linus war sogar stolz auf die Tatsache, dass er sich nie Bewegung verschaffte. Tove sah das offensichtlich anders. Ihre Karate-Pokale füllten ein ganzes Bücherregal, und auf dem Fernseher lagen ihre Aerobic-Videos. Das schien ein Streitpunkt zu sein. „Vielleicht wird mir in fünf Jahren irgendein Arzt sagen, dass ich abnehmen muss oder so“, sagte Linus damals.

Ich mache gern Sport und fand, dass Bewegung bei unseren gemeinsamen Unternehmungen eine wichtige Rolle spielen sollte. Ich wollte ihm Wellenreiten beibringen, aber es schien mir sinnvoll zu sein, erst einmal mit Boogie-Boarden zu beginnen. Wir fuhren an einem Nachmittag Anfang Mai nach Half Moon Bay hinüber, mieteten Trockenanzüge und Boards, und Linus protestierte heftig gegen den Gedanken, in die eisigen Wasser des Pazifiks zu waten, trotz des schützenden Trockenanzugs. Nach wenigen Minuten aber passierte etwas Erstaun-

liches: Es begeisterte ihn, in den Wellen zu reiten. „Das ist super“, sprudelte er wie ein Fünfjähriger und klatschte gegen meine Hand. Es kam, wie es kommen musste: Etwa fünfzehn Minuten später bekam er einen schlimmen Wadenkrampf - weil er völlig außer Form war, mutmaßte er - und musste aufhören. (Als ihn der Krampf überfiel, saß er einfach nur in der weißen Gischt, offenbar unfähig aufzustehen, und die Wellen spülten über ihn hinweg. Mein erster Gedanke war: „Oh Scheiße. Wenn ich zulasse, dass der Typ ertrinkt, bekomme ich es mit Millionen von Nerds zu tun.“)

Er freute sich auf alles, was wir zusammen unternahmen, während wir das Material für dieses Buch zusammentrugen: Tennis zu spielen, Wettschwimmen zu veranstalten, den schaurigen Amusements im Vergnügungspark Great America zu frönen, Golfbälle zu schlagen. Das ging schließlich so weit, dass er der Aktivität, die ich mir für uns ausgedacht hatte, mehr Interesse entgegenbrachte als dem Aufsprechen seiner Erinnerungen auf meinen Kassettenrecorder. Die Moorbäder, den Mount Tamalpais hoch zu wandern, Billard zu spielen, was auch immer. „Daran könnte ich mich gewöhnen“, sagte er heftig schwitzend nach einem Tennisspiel in der Nähe meiner Wohnung. Damals musste er sich Schläger und Laufschuhe noch von mir borgen. Später hatte er seine neuen Laufschuhe immer im Kofferraum seines Autos dabei, um für alle Fälle gerüstet zu sein.

Der Sinn des Lebens, 2

Haben Sie jemals in einer warmen Sommernacht auf dem Rücken gelegen, in die Sterne geschaut und sich gefragt, warum Sie eigentlich auf der Welt sind? Was Ihr Platz im großen Plan der Dinge und Ihre Aufgabe im Leben ist?

Tja, ich auch nicht.

Trotzdem habe ich mir irgendwann eine Theorie über das Leben, das Universum und all das zurechtgelegt. Oder jedenfalls über die Untermenge namens „Leben“. Die Einführung in diese Theorie haben Sie im Vorwort dieses Buches gelesen. Und jetzt, wo Sie schon so weit gekommen sind, würde ich sie gern noch etwas vertiefen.

Meine Theorie entstand jedoch nicht in einer klaren Nacht, in der ich in staunender Ergriffenheit über die Großartigkeit des Alls in die Sterne starrte. Sie entstand bei der Vorbereitung auf einen Vortrag. Wenn du für eine Sache bekannt bist, gehen die Leute wie selbstverständlich davon aus, dass du brillante Einsichten über dir völlig fremde Wissensbereiche aus dem Hut zaubern kannst, die seit Millionen von Jahren zu den ungelösten Rätseln der Menschheit gehören. Und sie erwarten, dass du diese Einsichten vor einer dir gänzlich unbekannten Zuhörerschaft ausbreitest.

Nein, das ergibt nicht allzu viel Sinn. Ich kam zu Linux, weil ich ein Technikfreak war, nicht weil öffentliche Auftritte oder das Philosophieren außerhalb der Grenzen meines Fachgebiets zu meinen Stärken gehören. Aber die wenigsten Dinge im Leben ergeben allzu viel Sinn, deshalb will ich mich nicht darüber beklagen.

Zurück zum Thema.

Dieses Mal war ich bei einer lokalen Veranstaltung in Berkeley eingeladen, die in Anlehnung an den Goldrush „Webrush“ hieß. Normalerweise hätte ich die Einladung ignoriert, aber ich bekam sie über das

finnische Konsulat hier in den USA, und weil ich ein Patriot bin (oder zumindest leichte Schuldgefühle habe, weil ich Schnee hasse und mich in wärmere Gefilde abgesetzt habe), sagte ich unbedacht: „Okay. *Jag gör det.*“*

Nein, niemand erwartete von mir, über den Sinn des Lebens zu sprechen. Ich am allerwenigsten. Aber bei der Veranstaltung ging es um das Leben in der vernetzten Gesellschaft, und ich war als Internet-Vertreter und finnischer Repräsentant eingeladen. In Finnland wird dank Nokia (dem größten, besten und schönsten Unternehmen der Welt, wie jeder Finne Ihnen bestätigen wird) im großen Stil kommuniziert, und das Stadium der „vernetzten Gesellschaft“ ist dort bereits erreicht. Sie wissen ja: In Finnland gibt es mehr Mobiltelefone als Menschen, und wir forschen derzeit daran, die Dinger gleich nach der Geburt chirurgisch zu implantieren.

So sitze ich also da und denke darüber nach, was ich zum Thema Kommunikation sagen könnte. Ach ja, ich habe ganz vergessen zu erwähnen, dass die anderen Teilnehmer auf dem Podium hauptsächlich Philosophen sein würden, die über Technologie reden würden. Schließlich waren wir in Berkeley. Und in Berkeley gibt es zwei heilige Kühe: Politikwissenschaften à la Berkeley und Philosophie à la Berkeley.

Was soll's. Wenn sie Philosophen über Technologie reden lassen, warum sollte dann nicht ein Technologietyp wie ich über Philosophie reden? Niemand sollte mir vorwerfen können, keinen Mumm zu haben. Sie mochten mich für strohduhm halten (was sie wahrscheinlich ohnehin taten), aber niemand sollte mich *feige* nennen.

Nicht diesen Geek.

Da sitze ich also und suche fieberhaft nach einem Thema, über das ich am nächsten Tag sprechen kann. (Ich bereite Vorträge immer erst auf den letzten Drücker vor, das heißt, normalerweise am Vorabend des Events.) Ich gebe mir wirklich alle Mühe und versuche, über die „Kommunikationsgesellschaft“ nachzudenken, und was sie für uns bedeutet, und wie Nokia und all die anderen Telekommunikationsgesellschaften sich wohl entwickeln werden.

Und mir fällt nichts Besseres ein, als einfach den Sinn des Lebens zu erklären.

* „Ja, klar, das mache

Nein, nicht eigentlich den „Sinn“ des Lebens. Eher das Gesetz des Lebens (im Folgenden „Linus' Gesetz“ genannt). Es ist äquivalent zum zweiten Hauptsatz der Thermodynamik der Physik, nur dass es nicht die Devolution der Ordnung im Universum erklärt, sondern die Evolution des Lebens.

Ich meine damit nicht „Evolution“ im Darwinschen Sinn. Das ist etwas anderes - für den Webrush-Vortrag interessierte mich mehr die gesellschaftliche Evolution und die Entwicklung von der Industrie- zur Kommunikationsgesellschaft: Was kommt als Nächstes, und warum? Ich wollte, dass meine Theorie gut klang, und genug Sinn ergab, um das Publikum für die Dauer einer Podiumsdiskussion zu überzeugen. Jeder hat seine ganz persönliche Tagesordnung, und meine sah für diesen Tag vor, lebend aus einer Podiumsdiskussion mit zwei hervorragenden Philosophen herauszukommen.

Warum also schreitet die gesellschaftliche Evolution voran? Was ist der Antriebsfaktor dafür? Treibt wirklich die Technologie die Gesellschaft voran - wie es der allgemeinen Ansicht zu entsprechen scheint? War die Erfindung der Dampfmaschine tatsächlich das auslösende Moment für die Entwicklung Europas zur Industriegesellschaft, aus der schließlich mit Hilfe von Nokia und Mobiltelefonen die Kommunikationsgesellschaft hervorging? Ich hatte den Eindruck, die Philosophen neigten dieser Ansicht zu und wollten wissen, wie die Technologie die Gesellschaft verändert.

Ich als Vertreter der technischen Seite weiß dagegen, dass Technologie gar nichts vorantreibt. Die Gesellschaft verändert die Technologie, nicht umgekehrt. Die Technologie legt nur die Grenzen dessen fest, was wir tun können - und wie billig wir es tun können.

Zumindest bis heute ist die Technologie ihrem Wesen nach ebenso unintelligent wie die Geräte, die sie hervorbringt. Das einzig Interessante an ihr sind die Möglichkeiten, die sie eröffnet, so dass die Antriebskräfte hinter jeder Technologie menschliche Bedürfnisse und Interessen sind. Wir kommunizieren heute nicht mehr als früher, weil wir die Möglichkeiten dafür haben - wir kommunizieren heute mehr, weil die Leute gerne quasseln und kommunizieren *wollen*, und wenn die Mittel dafür fehlen, werden sie eben geschaffen. Dafür gibt es Nokia.

Um zu wissen, in welche Richtung die Entwicklung der Gesellschaft fortschreitet, musst du deshalb wissen, so mein Argument, was die

Leute wirklich motiviert. Geld? Erfolg? Sex? Welcher Grundantrieb veranlasst die Leute, das zu tun, was sie tun?

Der erste offensichtliche Motivationsanreiz, den niemand bestreiten wird, ist einfach: das Überleben. Zu überleben ist schließlich die Grundbedeutung des Lebens. Leben gehorcht *nicht* einfach blind dem zweiten Hauptsatz der Thermodynamik, sondern überlebt trotz eines Universums, das der Komplexität und Ordnung, die das Leben für seine Absicherung braucht, ziemlich feindlich gesonnen zu sein scheint. Deshalb ist das Überleben der Motivationsfaktor Nummer eins.

Um die anderen Motivationsfaktoren in eine Reihenfolge zu bringen, musste ich überlegen, wie viel sie verglichen mit dem elementaren Überlebenswillen zählen. Die Frage lautet nicht „Würdest du für Geld töten?“, sondern „Wärest du bereit, für Geld zu *sterben*?“ Die Antwort ist ein klares Nein. Damit können wir „Geld“ eindeutig von der Liste der grundlegenden Motivationsfaktoren streichen.

Es gibt jedoch offensichtlich Dinge, für die Menschen bereit sind, ihr Leben zu opfern. Unzählige Heldengeschichten handeln von Menschen und sogar Tieren, die bereit sind, um einer größeren Sache willen zu sterben. Das schiere Überleben allein ist somit nicht der einzige Motivationsfaktor, der unsere Gesellschaft vorantreibt.

Die anderen Faktoren, die mir für den Vortrag in Berkeley einfielen, waren einfach und bei der Podiumsdiskussion kaum umstritten. Also sind zumindest einige Leute damit einverstanden. (Oder vielleicht waren sie auch einfach nur höflich - aus Rücksicht auf das finnische Konsulat.) Es gibt nicht viel, wofür die Menschen bereit sind zu sterben, aber soziale Beziehungen gehören definitiv dazu.

Zahlreiche Beispiele zeigen, dass eine soziale Motivation ausreicht, um Menschen das Überleben vergessen zu lassen: von Romeo und Julia in der Literatur (tot, nicht weil sie etwas Krudes wie *Sex* wollten, sondern weil sie lieber sterben wollten, als ihre Beziehung zueinander zu verlieren) bis hin zu dem patriotischen Soldaten, der bereit ist, sein Leben für Familie und Vaterland - sein gesellschaftliches Umfeld - aufs Spiel zu setzen. Halten wir also „soziale Beziehungen“ als Motivationsfaktor Nummer zwei fest.

Der dritte und letzte Motivationsfaktor heißt „Unterhaltung, Vergnügen, Spaß“. Das mag banal klingen, aber Spaß ist fraglos ein außerordentlich starker Antrieb. Jeden Tag sterben Leute, während sie Dinge tun, die sie nur zum Spaß machen. Dazu gehört zum Beispiel der

Sprung aus einem völlig funktionstüchtigen Flugzeug nur wegen des Kicks.

Unterhaltung braucht keineswegs banal zu sein. Es kann unterhaltsam sein, eine Partie Schach zu spielen. Es ist ein intellektuelles Vergnügen, herauszufinden, wie die Welt wirklich funktioniert. Es macht Spaß, seiner Neugierde nachzugeben und eine neue Welt zu entdecken. Und alles, was einen Menschen dazu bringt, in einer beengten Rakete auf einer Zillion Kilo hochexplosiven Materials zu sitzen, nur um sich die Erde vom Weltraum aus betrachten zu können, kann ganz sicher als „motivierend“ bezeichnet werden.

Und das war's dann auch schon. *Überleben. Dein Platz in der sozialen Rangordnung. Und Unterhaltung.* Diese drei motivieren unser Tun. Alles andere würde ein Soziologe vermutlich als „emergentes Verhalten“ bezeichnen - Verhaltensmuster, die sich auf jene sehr viel einfacheren Regeln zurückführen lassen.

Aber es steckt noch mehr dahinter. Die simple Aufzählung der Dinge, die die Leute motivieren, ergäbe keine richtige Theorie des Lebens. Das wirklich Interessante ist die Ordnung, die den drei Motivationsfaktoren innewohnt und die sich überall zeigt, wo es Leben gibt. Nicht nur *wir* werden von diesen drei Faktoren motiviert - sie gelten auch für nicht-menschliche Lebensformen, und jeder Lebensbereich entwickelt sich in dieser Reihenfolge.

Überleben. Sozialkontakte pflegen. Spaß haben. Das ist der Lauf der Dinge. Und deshalb wählten wir „Just for Fun“ als Titel dieses Buches. Weil alles, was wir tun, letztlich unserer eigenen Unterhaltung zu dienen scheint. Jedenfalls wenn wir die Chance erhalten haben, so weit zu kommen.

Sie glauben mir nicht?

Schauen Sie sich die Kriterien an, nach denen wir Tiere als niedriger oder höherentwickelt einklassifizieren. Sie alle überleben. Aber je weiter oben eine Tierart auf der evolutionären Skala steht, desto mehr bildet sie soziale Muster heraus - sogar Ameisen, die ziemlich weit unten auf der Skala stehen, kennen eine sehr strenge Sozialordnung - und irgendwann kommt Spaß ins Spiel. Ameisen neigen nicht sonderlich dazu, mit ihrem Futter herumzuspielen. Katzen schon. Ameisen haben auch keinen Spaß an Sex.

Ja, nehmen Sie etwas so Grundlegendes (und Schönes) wie Sex. Ich behaupte nicht, er gehöre zu den elementaren Motivationsfaktoren -

aber er ist ein sehr gutes Beispiel für ein ziemlich elementares menschliches Verhalten, das die höchste Evolutionsstufe erreicht hat. Sex war zunächst, das steht außer Frage, ein reines Überlebensverhalten. Schließlich haben sogar Pflanzen „Sex“ im Sinne des Überlebens, und in irgendeiner Phase vor Milliarden von Jahren war Sex auch für die einzelligen Tiere, aus denen eines Tages Geeks und andere Menschen hervorgehen sollten, eine reine Überlebensfrage.

Es steht auch außer Frage, dass sich Sex vor langer Zeit von der reinen Überlebensfrage zum gesellschaftlichen Phänomen entwickelte. Nicht nur beim Menschen ist Sex von Hochzeitszeremonien und einer ausgeprägten gesellschaftlichen Infrastruktur flankiert. Denken Sie an den rituellen Tanz der Sandhin-Kraniche, die sich übrigens für ihr ganzes Leben paaren. Tatsächlich wird jeden Tag unglaublich viel Energie für die Balzrituale verschwendet, die mit der einfachen Angelegenheit einhergehen, die Art zu reproduzieren.

Vergnügen? Das auch, das kann ich Ihnen sagen. Und nicht nur Menschen haben Spaß an Sex, auch wenn es vermutlich kein Zufall ist, dass die höchstentwickelte Spezies auf dem Planeten den größtmöglichen Unterhaltungswert daraus bezieht.

Der Fortschritt Überleben - Sozialverhalten - Unterhaltung ist in allen Bereichen zu finden. Nehmen Sie den Krieg: Damals, als du nur an das Wasserloch herankommen konntest, wenn du die Leute umbrachtest, die dir im Weg standen und die Wasserquelle für sich beanspruchten, war Krieg eindeutig eine Frage des Überlebens. Und später für lange Zeit ein Mittel, die Gesellschaftsordnung aufrechtzuerhalten. Und seit es CNN gibt, wurde er zur Unterhaltung. Ob es uns gefällt oder nicht: Das scheint der unausweichliche Verlauf zu sein.

Zivilisation und Kultur folgen dem gleichen Muster. Ursprünglich waren sie eine Möglichkeit, das Überleben durch Kooperation und zahlenmäßige Überlegenheit zu sichern. Das ist nicht nur beim Menschen so. Die meisten Tiere und sogar Pflanzen sind als Verbände organisiert, die sich gegenseitig helfen, um besser überleben zu können. Es ist interessant zu sehen, wie die Gesellschaft den Übergang vom reinen Überleben zum gesellschaftlichen Zusammenleben vollzieht: wie alle menschlichen Zivilisationen schließlich größere und bessere Straßen und Kommunikationskanäle bauen, um Sozialkontakte besser pflegen zu können.

In ihrer höchsten Form sind auch Zivilisation und Kultur dann immer stärker auf Unterhaltung ausgerichtet. Schauen Sie sich die alten Römer an: Noch mehr als für den Straßenbau und ihre starke Sozialordnung waren sie - vor allem in der späteren Phase - für Spaß und Spiele bekannt.

Oder nehmen wir die USA von heute. Zweifelt jemand daran, dass die Erfolge der Filmbranche und der Hersteller von Computerspielen nicht das Zeitalter der Unterhaltung einläuten? Was bis vor kurzem noch ein Nischenmarkt war, zählt heute zu den größten Industrien im reichsten Land der Welt.

Für mich als Technologe ist besonders interessant, dass dieses Muster sich auch in den Technologien wiederholt, die wir schaffen. Wir bezeichnen das frühe Zeitalter der modernen Technologie als Industriezeitalter, aber eigentlich sollten wir es das Zeitalter des technologischen Überlebens nennen. Technologie diente bis vor kurzem fast ausschließlich dem Überleben: Mit ihrer Hilfe war es möglich, Stoffe besser zu weben und Güter schneller zu transportieren. Das war ursprünglich der eigentliche Antrieb.

Die aktuelle Phase bezeichnen wir als Informationszeitalter. Das bedeutet eine enorme Verlagerung. Technologie unterstützt jetzt die Kommunikation und Verbreitung von Informationen - also ein überaus soziales Verhalten - und nicht mehr nur ein besseres und angenehmeres Überleben. Das Internet und die Tatsache, dass ein so großer Teil unserer Technologie sich zunehmend in diese Richtung bewegt, ist ein unmissverständliches Zeichen: Es bedeutet, dass die Leute in den industrialisierten Ländern das Überleben als selbstverständlich betrachten und plötzlich die nächste Phase der Technologie neu und aufregend finden. Der soziale Aspekt der Kommunikationstechnologie rückt in den Mittelpunkt; Technologie wird nicht nur eingesetzt, um besser leben zu können, sondern entwickelt sich zum festen Bestandteil des gesellschaftlichen Lebens.

Der Gipfel dieser Entwicklung ragt natürlich noch vor uns auf. Über die Informationsgesellschaft hin zur Unterhaltungsgesellschaft. Eine Welt, in der das Internet und drahtlose Kommunikation rund um die Uhr als selbstverständlich gelten und keine Schlagzeilen mehr machen. Eine Zeit, in der Cisco zur Old Economy gehört, und die Disney Corporation die Welt beherrscht. Eine Zeit, die nicht mehr allzu fern sein dürfte.

Was hat das alles zu bedeuten? Wahrscheinlich nicht viel. Schließlich ist meine Theorie über den Sinn des Lebens keine *Anleitung* für Ihre persönliche Lebensführung. Sie können daraus bestenfalls entnehmen: „Ja, du kannst dich dagegen wehren, aber das höchste Ziel im Leben ist und bleibt es, Spaß zu haben.“

Bis zu einem gewissen Grad erklärt das, warum die Leute so bereitwillig und begeistert an Projekten wie Linux mitarbeiten, die durch das Internet möglich wurden. Linux bot mir und vielen anderen Menschen eine Möglichkeit, gleichzeitig *zwei* Motivationsreizen, die uns juckten, nachzugeben. Als das Überleben zur Selbstverständlichkeit geworden war, brachte Linux seinen Fans sowohl das Vergnügen an der intellektuellen Herausforderung als auch die soziale Motivation, die damit verbunden ist, Anteil an seiner Entwicklung zu haben. Wir haben vielleicht nicht viel voneinander gesehen, aber die E-Mails waren viel mehr als nur ein trockener Informationsaustausch - per Mail entstehen Freundschaften und andere soziale Bindungen.

Und noch etwas lässt sich daraus ableiten: Falls und wenn wir jemals anderen intelligenten Lebewesen in diesem Universum begegnen sollten, werden ihre ersten Worte wahrscheinlich nicht lauten: „Führt mich zu eurem Oberhaupt.“ Sondern wahrscheinlich werden sie sagen: „Macht weiter Party, Leute!“

Aber vielleicht liege ich damit ja falsch.

Register

486er DX266 128
68008-Chip 49f., 59

A

Adams, Douglas 36
Advocacy-Newsgroups 167, 171,
174
AIX-Betriebssystem 169
Allen, Paul 152
Alpha-Computer 142
Amiga-Computer **47, 95, 110,**
142
Amoeba-Betriebssystem 107, 110,
112, 114
Andrew Tanenbaum
 Operating Systems: Design and
 Implementation 60, 62, 121
Anti-Kommerzialismus der Linux
 Anhänger 170, 176
Anvin, Peter 71, 128, 153, 155
Apache (Linux für Web-Server)
 169f., 179
Apple Computer 160, 162
Apple-II-Computer 47
Assemblersprache 47, 49, 52ff., 70
Asteroids (Computerspiel) 54
AT&T 61, 63 ff., 237
Atari 110
Augustin, Larry 185f.
Avuton 255

B

Balmer, Steve 180
BASIC (Programmiersprache) 15,
18, 47, 49
Behlendorf, Brian 179
Bell Labs (AT&T) 61, 63
Berinato, Scott 198, 200
Berkeley Software Distribution 64
Berner Konvention 103
Betriebssysteme
 Entwicklung von 83
 Microkernel-Ansatz 107f., 110,
 113, 161
 monolithischer Ansatz 110,
 113
 Torvalds' erste Basteleien 52
Binärdateien 90, 104, 110
BIOS 69
BMW Z3 117f., 214, 252,
253
Boies, David 199
Börsengang
 Red Hat 184, 186
 VA Linux 186f., 189
Bourne-Shell (Unix) 91
Brecht, Bertolt 41
Browser, Netscape 167f., 248
Brute-Force-Methode 84
BSD 64
Bündelung von Software 233

- C
C (Programmiersprache) 53, 60, 95
Cabal, Internet 126 f.
Caldera 140
Carnegie Mellon University 160
CDs 231 f.
China 106, 245
- Cisco 264
CNN 109, 111, 186
Coherent (Betriebssystem) 101, 139
Comdex (Computermesse) 5, 191 f., 245
Commodore 64 47
Commodore VIC-20 14
Common Desktop Environment 140
Compaq 217
Computer
 fingern 204
 Kostenentwicklung SOF. und Mathematik 82
 und Physik 81
Computerspiele 3, 24, 54f., 64
Copyleft 105
Credit List 131
- D
D'Cruze, Patrick 145
Dateisystemtreiber 85
Datenbanken, und Linux 171
DeCSS-Prozess 228
DECUS (Usergroup von Digital) 142
Dell Computer 217
Diamond, David
 Boogie-Boarden mit Torvalds 58, 256
 Comdex-Vortrag 195ff.
 erste Begegnung mit Torvalds 27ff., 31
 Interviews mit Torvalds' Familie 72 ff., 79
- Pinguin als Linux-Symbol 149, 151
Torvalds' BMW Z3 117
 Erinnerungen an die Anfangstage von Linux 115 ff.
 finanzielle Ängste 188 f.
 Kindheitserinnerungen 43 f.
 neues Haus 214
 Telefonnummer 183
 Theorie über den Sinn des Lebens 1 ff., 7
 Veränderung 255ff.
Diamond, Kaley 149
Digital Equipment Corporation 61, 142 f., 154
Digital-Millennium-Copyright Gesetz 222 f., 227
Disassembler 52
Disney Corporation 264
DOS-Betriebssystem 47, 62, 68, 80, 110, 173
DVD-Technologie 228, 232
- E
Einstein, Albert 224, 251
Evans, Bruce 69, 96, 109, 111 f.
Evolution, Torvalds' Theorie 165, 210, 237, 260, 263
Ewing, Larry 151
- F
Farfar 21
Farmor 21
Fermilabs 173
Festplattentreiber 85
Fingern von Computern 204
Finnish News Agency 12, 23
Finnland
 Ball des Präsidenten 211 ff.
 Erfolgsneid 139
 Geschichte 20
 Handynutzer in 41
- Rentiere 40f. Saunas in 40f.
Schulsystem 32ff., 48 Sommer in 91 Stoizismus 41
Unabhängigkeitstag 211 und neue Technologien 42 Winter in 17 First Boston Credit Suisse 186 Forbes (Zeitschrift) 174 fork (Unix-Operation) 62 Forth (Programmiersprache) 51 Freax 92 Free Software Foundation 66, 178 Free-Software-Manifest 66 FreeBSD 65
- G
Gates, Bill 27, 30, 179, 191 ff., 196, 199, 243
 Comdex-Vortrag 1999 191, 194 öffentliches Image 173, 179 Torvalds verglichen mit 27, 30, 179 und
 Open-Source-Philosophie 244 Gauss, Carl Friedrich 84
GCC-Compiler 96, 98, 105 Geeks 12, 34, 83, 85, 87, 186, 259
geistiges Eigentum 219, 221 ff.
Gedanke des fairen Gebrauchs 223, 227
Geschäftsgeheimnisse 224
Kreativität 220, 226 Patente 223 und Internet 228 Urheberrecht 220f., 223 wirtschaftliche Interessen 227 General Public License 105
Geschäftsgeheimnisse 223 ff., 228
Gesellschaftsordnung
- als Motivationsfaktor 2ff., 6, 261, 263 Gesetz der Entropie 6 Gier 103, 106, 176, 230, 232, 235
GNU-Betriebssystem 66, 94f., 112f., 175, 179, 208 GNU-Emacs 86 Gnutella 231 GPL 66, 106, 168, 178, 207, 208f. Grafik, Computer SS grafische Benutzerschnittstelle 124 Griechen, Gesellschaftsordnung 6
GSM-Standard, Telekommunikation 230
- H
Hall, Henry 150
Hall, Jon (Maddog) 142, 145f., 150, 154, 196
Halloween Document (Microsoft-Memo) 179 Handspring 234
Hardware-Emulation 158
Helsinki Telephone Company 25
Henry, Bischof 20 Hewlett Packard 234 History File 131
Hohndel, Dirk 184f.
- I
IBM 56, 160, 169ff., 217, 234, 241
Industriegesellschaft 260
Industriezeitalter 264
Informationszeitalter 264
Informix 171
Intel 110, 143, 147, 153, 158, 178
- J
Java (Programmiersprache) 162, 168, 170, 233ff.
Jini (Sprache für interagierende Agenten) 162f., 169

Jobs, Steven 160 f., 165
Jolitz, Bill 64
Joy, Bill 64, 162ff.

K

Kommerzialismus, Sorgen der Linux
Anwender 152, 170, 176
Kommunikationsgesellschaft 259f.
Kommunismus und Torwalds'
Vater 22, 73, 103
Kontrolle von Ressourcen 230ff.
Softwarebranche 233 f.
Unterhaltungsbranche 231 ff.
Kreativität des Programmierens 82
ff. und geistiges Eigentum 220,
222, 226
Krieg, Torwalds' Theorie über den
Sinn des Lebens 3, 6, 263

L

Lemke, Ari 88f., 96f.
Linux 175 als Bedrohung für
Microsoft 180 als globales
Gemeinschaftsprojekt 131 an den
Hochschulen 173 Angst vor
Zersplitterung 181
Anwenderunterstützung 217
Copyright-Bestimmungen 103,
105 eingebettete Systeme 173
Entwicklungsmodell 130 erste
Anfänge 85f., 88 erste Portierung
142 Fan-Websites 139 Feedback
in der Anfangsphase 97, 116, 120
grafische Benutzerschnittstelle
124 f. Idealisten gegen
Pragmatiker 175

Kein-Geld-Philosophie 102, 104
kommerzieller Erfolg 124, 172,
216, 241 Newsgroups 107,
126f., 152f. PR-Aktivitäten
137 Sicherheitsfragen 126
Szene 101 f., 104 und die
Technologie der
Zukunft 239 und
konkurrierende
Betriebssysteme 172
ursprünglich geplanter Name 92,
97 verglichen mit Microsoft 173
Vernetzung 125 f. Version 0.01 96
Version 0.02 99 Version 0.10 100
Version 0.12 101 Version 0.95 125
Version 1.0 137 Version 1.2 143
Vorzüge 172 wachsende Beliebtheit
126 Warenzeichen 145 f.
Warenzeichenstreit 144 Linux
International 142, 145f., 169 Linux
Journal 143 ff. Linux SMP 157
Linuxcare 217 LISP-Maschine 207
Looking Glass (Unix-Desktop) 140

M

Mac OS X-Betriebssystem 160
Mac-Betriebssystem 89
Mach-Betriebssystem 110, 160ff.
Maddog 142
Mainframe-Computer 172
Mandelbrot, Benoit 83
Mandelbrot-Mengen 83
Mark Williams Company 101

Mathematik 15, 35, 82, 84 McKusik,
Marshall Kirk 64, 123
Microkernel-Ansatz 107, 110f., 161
Microsoft 173, 239
Antitrust-Verfahren 179, 192, 194
Bedrohung durch Linux 179
Bündelung von Software 233
Halloween Document 179 und Sun
Microsystems 234 MicroVAX 61 f.
Minix-Betriebssystem 60, 68ff.,
85ff., 92, 94ff., 99 ff., 107f., 110 f.,
113 f., 139 als
Linux-Entwicklungsumgebung 99
gegen Linux 107f., 110ff., 114f.
Lizenzpolitik 109 Newsgroups 107
Torwalds als Anwender 60, 68 f., 86
und Multithreading 112 MIT 124,
207 Mobiltelefone 4, 41f., 259f.
Modems 70, 101, 125 monolithischer
Ansatz 110, 113 Morfar 17
Motivationsfaktoren 261
Open-Source-Programmierer 131,
243 Torwalds' Theorie über den Sinn
des Lebens 2, 4 f., 261 f., 264 f.
Motorola 49, 142 Mozart, Wolfgang
Amadeus 226 Mozilla
(Netscape-Projekt) 167, 169, 248
Mozilla Crypto Group 169
Multitasking 49, 112, 140
Musikindustrie 106, 231 f.

N

Napoleon 20
Napster 231
NASA 173
Nesson, Charles 242
NetBSD 65
Netscape 27, 167ff., 247
Newsgroups
Advocacy 167, 171
Linux 126, 152, 156
Minix 69
Newton, Isaac 103
Nokia 42, 103, 259f.
Norssen High School (Helsinki)
32 f.
Norwegen 41
Novell 65, 140

O

Open Law Project 242
Open Source
Apple Computer 161
Einstellung zu Fragen des geistigen
Eigentums 225 Fanatismus
208 ff. IBM 169 f. im
Rechtswesen 242 Informix 171
Linux' Bedeutung für 254
Motivation zur Mitarbeit 131,
243 Netscape 168 Oracle
171 Philosophie 152, 173,
241 f. Sun Microsystems
163, 169 Szene 167, 207 und
die freie
Softwarebewegung 178
und Wirtschaft 176
Vorteile 207f., 233, 241,
244
OpenBSD 65

Open-Source-Projekte
 Netscape 247
 Wapit 246f. '
 Operating Systems: Design and Implementation (Tanenbaum) 60, 121, 132
 Oracle 171
 OS/2-Betriebssystem 170

 P Pacman (Computerspiel) 54f.
 Page-to-Disk 100ff. Paging 56, 95
 Palm 234 Patches 69, 109, 181, 248
 Patente 200, 219, 223f. PC Week 198 PDP,-11-Computer 64 Pentium FO OF-Fehler 178 Personal Computer 47, 49f., 56, 61, 68 f., 71, 102, 142, 170, 238 f., 241 Pinguin (als Linux-Symbol) 149ff. Pipes 63, 70 Playboy 23, 25 Portabilität 110, 113 POSIX-Standards 86ff., 113f. Prinz von Persien (Computerspiel) 68 Programmierung Torvalds' erste Schritte 15 f. und Kreativität 81 f., 84 Protected Mode 69 Protest gegen das Establishment 173

Q
 Q-DOS-Betriebssystem 49

R
 Randi (Torvalds' Katze) 59, 132f., Ravmond. Eric 168. 175. 179. 209

Real Mode 69 Red Hat 154, 163, 170f., 176, 178, 185, 187, 205, 217
 Red Herring (Zeitschrift) 72
 Rentiere (Finnland) 40f. Ritchie, Dennis 63 ROM (Read-only Memory) 52, 69 Römer, alte 264
 Root Beer, Torvalds' Abneigung gegen 141, 147
 Root-Benutzererkennung 204
 Russland 20, 22, 41 f., 72

S San Jose Mercury News 27 Saunas 28f., 40ff., 118, 141 Saxberg, Ole Torvald Elis (Großvater) 21
 Schweden 20, 76, 122, 153
 Scientology 229 Sex, Torvalds' Theorie über den Sinn des Lebens 2f., 261, 263 Shareware-Tradition 102 Shell-Kommando 62 Sibelius, Jean 188 Sicherheitheitsfragen, und Linux 126 Silicon Graphics 217
 Silicon Valley 27ff. 152, 157, 160, 164 f., 18 7, 214 Sinclair QL (Computer) 51, 53f., 56ff., 61
 Eigenschaften 49 Finanzierung 49f. Geschichte 49, 56 Microdrive 51, 57
 Sinclair, Clive 49, 56
 Softwarepatente 223 Sony 238f.
 Space Wars (Computerspiel) 64
 Speicherverwaltung 56f., 89, 110
 Spektrum 35, 85, 119, 139

Spice Girls 226
 Stalluran, Richard 66, 105, 175, 178f., 207
 Star Office 164, 193
 Sun Microsystems 87, 90, 143, 162 ff., 168 f., 233 ff.
 SuSe (Linux-Version) 171
 Symbolics 208
 Systemaufrufe 62f., 87, 89ff.

T
 Taligent (Betriebssystem) 160
 Tanenbaum, Andrew Angriff gegen Linux 107ff. und Minix 69
 Taskwechsel 70, 95
 TCP/IP (Vernetzungsstandard) 126
 Technische Universität von Helsinki 66, 88
 Technologie künftige Entwicklungen 237, 239f. und Torvalds' Theorie über den Sinn des Lebens 3 f., 6, 260, 264
 Tele 154
 Telekommunikation, Standards 230
 Terminal-Emulation 69, 85, 89, 124
 Terminal-Emulator 99
 Tevanian, Avie 160f.
 Thompson, Ken 63
 Threads 70
 Timex-Computer 49
 Tischrechner, elektronisch 14f.
 TiVo 245
 Törnqvist, Leo Waldemar 14f., 19, 21, 44
 Torvalds, Anna (Mikke, Mutter) 22f., 25, 75ff., 79
 Torvalds, Daniela (Tochter) 1, 5, 43, 167, 189, 193, 214f.

Torvalds, Leo (Stiefbruder) 18, 23
 Torvalds, Linus
 Aktienoptionen 178, 184ff.
 als Forschungsassistent 124
 als Inhaber des Linux-Warenzeichens 146, 177 als Lehrassistent 124, 132 als Linux-Oberhaupt 129f., 180 als Trend-Guru 236 Angst vor Copyright-Verletzungen 105 Aussehen als Kind 11
 Äußerungen über Bill Gates 30
 Begegnung mit Bill Joy 162 f.
 Begegnung mit Steve Jobs 160
 Betrachtungen über Evolution 237, 260
 Comdex-Vortrag 191 ff.
 Computer als Hobby in der Kindheit 14ff.
 Entscheidungsfreiheit 209 erste Programmiersuche 16, 18 erste USA-Reise 140 f.
 erster PC 67ff. familiärer Hintergrund 20 ff. Ferienjobs 48
 Freigabe von Linux 0.01 96
 Freigabe von Version 1.0 137f.
 Gedanken über Genetik 165
 Heirat 155 Job bei Transmeta 157f., 181 Journalisten und 22f., 139, 180, 182, 201, 212, 214, 252
 Katzen 34, 59, 133, 155
 Linux-ist-nicht-verkäuflich-Einstellung 104 öffentliches Image 27, 127, 173, 179f., 186, 202, 252
 öffentliches Sprechen 122 f., 182, 205

Online-Sammlung für ersten PC 71, 128, 153
 Presseinterview bei der Linux World Expo 199
 Schlafgewohnheiten 159
 Schlagabtausch mit Andrew Tanenbaum **111**, **113** f., 120
 Schönheit des Programmierens 81 ff.
 Schulstipendien 48
 Schulzeit 25, 32 ff.
 Sinn des Lebens 258
 soziale Unsicherheit 3 5, 85, 119
 Sport 58, 116, 171, 256
 Studium 35, 42, 58, 148
 Sushi mit Billjoy 164f.
 Terminal-Emulationsprogramm 69 ff.
 Theorie über den Sinn des Lebens 1 ff., 7, 258 f., 261 ff.
 Umzug nach USA 156
 verglichen mit Bill Gates 28, 173, 179
 Verhältnis zu Geld 25, 28, 102f., 253
 Vor- und Nachteile des Ruhms 251 ff.
 Vorstellungsgespräch bei Transmeta 153
 Wehrdienst 22, 36ff., 56, 58f., 236
 Zugänglichkeit 198ff.
 Torvalds, Märta (Farmor) 21
 Torvalds, Nils (Nicke, Vater) 18, 22, 24 f., 72 f., 75 ff., 214
 Torvalds, Patricia (Tochter) 1 f., 5, 43, 79, 142, 155f., 166f., 189, 193, 214f.
 Torvalds, Sara (Schwester) 12, 18, 22ff., 33, 70ff., 74ff., 92, 101, 103, 167

Torvalds, Tove (Ehefrau) 1, 19, 31, 43, 58, 78, 116, 132f., 138f., 142, 144f., 150, 152, 155f., 171, 182, 184f., 212, 214, 255f. Ball des Präsidenten 211 Beziehung und Ehe 132f., 150, 155
 Comdex-Vortrag 192f. Geburten der Töchter 152, 155 Journalisten 139, 150 neues Haus 214 Pinguin als Linux-Symbol 149 Umzug in die USA 147, 156
 Transmeta Corporation 27f., 79, 117, 124, 152f., 156ff., 171, 178f., 181, 190, 193, 196, 198, 200 f., 245
 Transmeta Web Slate 245
 U
 Überleben, als Motivationsfaktor 2ff., 6, 237, 243, 261 ff.
 Ultrix-Betriebssystem 61
 United States Copyright Extension Act 242
 Universität Helsinki 14, 35, 56, 61 f., 126, 138, 152, 154
 Universität Illinois 168
 University of California, Berkeley 64
 Unix Domain Sockets 125
 Unix-Betriebssystem 27, 60ff., 68f., 87, 89f., 96, 101, 104, 117, 122f., 125, 140, 142, 163 f., 169, 172, 179, 181 f., 204 Anfänge 64
 Anfangstage 164 Bourne-Shell 91
 Einfachheit 64
 Entwurfsphilosophie 63
 konkurrierende Versionen 64

Operationen 62 Philosophie 65
 Pipes 63 Prozessbegriff 62
 Shell-Kommando 62 Szene 65
 Unix-Szene 65 Zersplitterung 181
 Unterhaltung 6
 Unterhaltungsindustrie 228
 Urheberrecht 103, 105f., 208, 221, 223, 225, 231 Berner Konvention 103 geistiges Eigentum 220ff. Linux 103 ff. Musikbranche 226
 Überlegungen zum 105 f.

V VA Linux 146, 186f., 191, 244
 Verbraucherschutz, und
 Urheberrecht 222 Vergnügen, als Motivationsfaktor 2 ff., 261 f.
 Vernetzung 125 f., 162 VIC-20 15, 44, 47, 50, 54 Vierumaki, Jouko 57, 76, 80 Visor 234
 VMS-Betriebssystem 61, 63, 110

W Wahrnehmung, Rolle von 238
 Wapit 246 Warenzeichen 27, 144ff., 223 Warenzeichenstreit 144
 Webrush (Veranstaltung der Universität Berkeley) 258 Webserver, Linux und 172, 241 Wendt, Ernst von (Urgroßvater) 22 Wikinger 20
 Windows (Betriebssystem) 63, 127, 170 Windows NT 172, 180 Wired (Zeitschrift) 165 Wirzenius, Lars 35, 66, 71, 117, 124, 137 Wissenschaft, als Bedrohung des Establishments 244 Wosniak, Steve 49

X x86er Interpreter 158
 XENIX-Betriebssystem 1
 X-Window-System 124f. 17

Z
 Zapfsäule und Linux 245
 Zborowski, Orest 125