

IV.

2. 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 18000 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. heraussuchen 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 namen-, 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 Volvo-Kombis 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 Uni-Computer 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 vergraule ich 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, AAAAAAAA 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 abbezahlt 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.

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 willkom-
men, aber ich verspreche nicht, dass ich sie imple-
mentieren werde :-)
```

Linus (torvalds@kruuna.helsinki.fi)

PS. Ja - es ist frei von Minix-Code und hat ein Multi-
threading-Dateisystem. Es ist nicht portierbar (ver-
wendet 386er Taskwechsel usw.) und wird
wahrscheinlich nie etwas anderes als AT-Festplatten

unterstützen, weil das alles ist, was ich zur Verfügung habe :-).

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

>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 Kernel 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 GCC-Compiler 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 selbständig 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 logging 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. Virtueller. 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-to-Disk 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 legten. Mit Page-to-Disk konntest du größere Programme laufen lassen, als in deinen Speicher passten. Wenn der Speicher knapp wurde, konntest du alte Hauptspeicherinhalte 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-eine-Postkarte. In der PC-Welt von damals gab es eine starke Shareware-Tradition. 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 rüberzuschicken. 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-Handies 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 Kernel 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.