

Uptime

Mitgliederzeitschrift der
German Unix User Group



Ghostscript-Kung-Foo mit inkcov

RFC 2822: Spaß mit E-Mail-Adressen

Bilder-Ernte vom GUUG-Empfang

guug

2013-2

Inhaltsverzeichnis

Liebe Mitglieder, liebe Leser! <i>von Wolfgang Stief, Vorsitzender des Vorstands</i>	3
Vereinsleben <i>von Anika Kehrer</i>	5
In Gedenken an Ulrich Gräf <i>von Volker A. Brandt</i>	7
PDF-Kung-Foo mit Ghostscript <i>von Kurt Pfeifle</i>	10
Wenn der Mini-GAU kommt <i>von Stefan Schumacher</i>	17
Shellskripte mit Aha-Effekt <i>von Jürgen Plate</i>	22
Geschichtsstunde III <i>von Jürgen Plate</i>	24
#ffg2013 <i>von Corina Pahrman</i>	31
Bildernachlese GUUG-Empfang 2013 <i>von Anika Kehrer</i>	34
Himmel hilf! <i>eine Schmähschrift von Snoopy</i>	38
Hilfreiches für alle Beteiligten	43
Über die GUUG	45
Impressum	46
GUUG-Mitgliedsantrag	47

Liebe Mitglieder, liebe Leser! Grußwort vom Vorstand

Nach dem FFG ist eine Verschnaufpause und ein zufriedener Rückblick erlaubt. Als Lektüre für die Sommerferien viel Spaß mit der neuen redaktionellen Uptimes!

von Wolfgang Stief, Vorsitzender des Vorstands

Hi.
Wo?

Ein lustiger Mensch.

Diese Uptimes ist nicht die erste in diesem Jahr: Die erste Ausgabe 2013-1 blieb den Proceedings zum Frühjahrsfachgespräch (FFG) in Frankfurt vorbehalten, die schon geraume Zeit unter [1] zum Download bereit steht. Weil die Proceedings – noch – etwas anders produziert werden als die anderen Uptimes', gibt es die Proceedings nur als PDF, nicht auch als ePub.

Für die GUUG begann das Jahr mit einem erfolgreichen FFG in Frankfurt, den der Artikel „#ffg2013“ in diesem Heft noch einmal Revue passieren lässt. Der neu eingeführte dritte Vortragstrack am Donnerstag wurde gut angenommen. Danke an dieser Stelle im Namen des Vorstands an alle Sprecher, die Vorträge eingereicht haben! Wenn es in 2014 wieder eine solch große Zahl hochwertiger Einreichungen gibt, werden wir wohl auch dort wieder einen dritten Track gestalten. Interessante Themen gibt es genug. :-)

Traditionell findet am Abend vor dem ersten Konferenztag des Frühjahrsfachgesprächs die Mitgliederversammlung der GUUG statt. Weil einige Punkte erwartungsgemäß etwas hitziger und ausführlicher diskutiert wurden, wurde am Ende die Zeit eng: Der Wachschutz der Hochschule hat uns kurz nach 22 Uhr gebeten, den Raum zu verlassen, und wir mussten die Versammlung abrechnen, ohne ein Budget für 2013 genehmigt zu haben. Ende Mai – am Rande des LinuxTags in Berlin – fand eine weitere Mitgliederversammlung statt, in der dann auch das Budget genehmigt wurde. Mangels genehmigten Budgets konnten wir in der ersten Jahreshälfte nicht besonders aktiv sein, weshalb es

Links

[1] Proceedings des FFG 2013 in Frankfurt: <http://www.guug.de/uptimes/2013-1/>

davon auch nichts weiter zu berichten gibt. Ein paar Details findet Ihr in dem Artikel „Aus MV und Vorstand“ trotzdem in dieser Ausgabe.

Was wir uns nicht haben nehmen lassen: den traditionellen GUUG-Empfang anlässlich des LinuxTags in der Alten Pumpe. Dieses Jahr leider aufgrund der Witterung nicht draußen im Biergarten, sondern drinnen. Was der Stimmung keinen Abbruch tat, wie die „Bildernachlese GUUG-Empfang 2013“ weiter hinten in dieser Ausgabe zeigt.

Für den Rest 2013 (und sicher auch die ersten Monate 2014) haben wir jedoch einige schöne Ideen, von denen sich hoffentlich die meisten auch umsetzen lassen. Die Mithilfe und Mitarbeit der Mitglieder steht dabei weit vorn. Mehr demnächst über den Newsletter oder als Diskussion auf <guug-members@guug.de>!

Überschattet wurde die Produktion dieser Uptimes von dem plötzlichen und unerwarteten Tod von Uli Gräf. Uli war gern gesehener Sprecher am FFG und auf der OSDevCon, immer mit Themen aus der Solaris-Welt. Volker Brandt hat kurzfristig einen Nachruf für die Uptimes verfasst, was wir sehr schätzen und wofür wir ihm danken. Das gilt auch für diejenigen, die die schönen Bilder beige-steuert haben.

Ich wünsche allen Lesern eine interessante Lektüre und freue mich schon auf die zweite redaktionelle Ausgabe im Winter dieses Jahres. Wer Fragen dazu hat oder mit dem Gedanken an einen Artikel spielt, ist jederzeit bei <redaktion@uptimes.de> goldrichtig.

Über Wolfgang



Wolfgang Stief hat Linux während des E-Technik-Studiums kennen und schätzen gelernt (Kernel 0.99). Seit 1998 verdient er sein Geld überwiegend mit Solaris, Storage-Systemen und verteilten Dateisystemen. Wenn er gerade mal nicht beruflich oder für die GUUG unterwegs ist, züchtet er Gemüse, treibt sich in sozialen Netzwerken herum oder macht Musik.

RECENTLY IN THE COMPUTER MUSEUM



CIRCLE OF TIME

Vereinsleben Aus Verein und Vorstand Anfang bis Mitte 2013

Die GUUG ist ein lebendiger Verein mit einer jährlichen Konferenz, aber auch Mitgliederversammlungen und mehreren Vorstandssitzungen pro Jahr. Dieser Artikel schließt diejenigen an, die weniger am Geschehen teilnehmen.

von Anika Kehrer

Reden ist Silber,
Handeln ist Gold.
Gemeinsam handeln ist Dynamit.

Frei nach dem bekannten Sprichwort unbekannter Herkunft.

Das Frühjahrsfachgespräch der GUUG – wovon weiter hinten im Heft noch zu lesen ist – hat wieder die Mitgliederversammlung (MV) mit sich gebracht, am 27. Februar in Frankfurt. Anwesend waren 54 GUUGler (im Vergleich zu 50 im Jahr 2012 und rund 40 im Jahr 2010). Die Mitgliederzahl der GUUG, wie auf der MV zu lernen war, lag zum Ende des Jahres 2012 bei 655 (2010: 661, 2011: 678). Die Tendenz bei regulären Mitgliedern ist steigend, während sie bei Fördermitgliedern sinkt und bei ermäßigten Mitgliedschaften eingebrochen ist (von 39 Ermäßigten in 2011 auf 22 in 2012).

Schichtwechsel im Vorstand

Dieses Jahr hat der Verein seinen Vorstand neu gewählt: Als Vorsitzender wird Wolfgang Stief den Verein zwei weitere Jahre vertreten. Seine Stellvertreter sind Kerstin Mende-Stief und Christoph Wegener. Als Schatzmeister bleibt Udo Arendt im Amt. Der nunmehr sechsköpfige Vorstand wird komplettiert durch die Beisitzer Johannes Hubertz und Bernd Neubacher. Dieses Team führt nun das vor etwa zwei Jahren begonnene Bestreben weiter, die GUUG in der Öffentlichkeit sichtbarer zu machen.

Eine Maßnahme davon waren Beteiligungen und Sponsorschaften bei Konferenzen und Messen. In diese Maßnahmen-Ecke sind, wie der alte Vorstand noch auf der MV berichtete, insgesamt etwas weniger als 10.000 Euro geflossen. Am meisten profitierte das Team der Berliner EHSM (Exceptional Hard and Soft Meeting) vom Engagement der GUUG, die Unterstützung in Höhe von 3.000 Euro erhielten. An zweiter Stelle folgt die FrOS-Con in St. Augustin bei Bonn: Für einen eigenen Stand und Sponsoring nahm die GUUG hier etwas über 2.000 Euro in die Hand. Mit eigenem Stand

sowie Sponsorenlabels war der Verein weiterhin auf der SIGINT in Köln (1.190 Euro) und auf dem Linuxtag Augsburg vertreten (800 Euro). Flagge zeigte die GUUG als Sponsor schließlich bei der DENOG in Darmstadt (1.100 Euro) und bei den Chemnitzer Linux-Tagen (1.150 Euro).

Zukunftswerkstatt GUUG e.V.

Die zweite Linie, an der der GUUG-Vorstand arbeitet, ist ein lebendiges Vereinsleben. Dafür war im Herbst 2011 die Entscheidung gefallen, die interne Kommunikation des Vorstands – und damit die Vorstandsarbeit – auch für GUUG-Mitglieder zu öffnen, die nicht die Funktion eines gewählten Vorstandes besitzen [1]. Auf diese Weise sind zum Beispiel auf der ersten Vorstandssitzung dieses Jahres, am 9. Februar 2013 in Bochum, sowie auf der zweiten am 21. Mai in Berlin im Rahmen eines gemeinsamen Abendessens am Vortag der MV sechs GUUG-Mitglieder Subskribenden geworden. Derzeit sind neben den sechs Vorständen insgesamt 13 Mitglieder eingetragen – und diskutieren auch mit.

Auf dem Fuße folgt – wie das so ist – eine kleine Komplikation. Eigentlich, merkten die Vorstände, dürfte die Aufnahme in den Vorstandsvertreiter nur so lange laufen, wie der beschließende Vorstand im Amt ist. Bei Wechsel im Vorstand müsse neu abgestimmt werden. Das ist ein Beispiel für formale Details, die in der Vereinsarbeit auch bedacht sein wollen.

In etwas größerem Rahmen, nämlich sowohl auf Mailinglisten als auch auf beiden Mitgliederversammlungen, ist die Mitgliedschaft in der Bitkom gerade ein Thema (die zweite MV fand mit etwa 25 Mitgliedern im Rahmen des LinuxTag statt). Einerseits, so verläuft die Debatte, bestehe

darin die Chance, das Vereinsanliegen – Förderung offener Systeme – zum Beispiel im Rahmen der Open-Source-Arbeitsgruppe zu fördern. Andererseits jedoch wird die Mitgliedschaft als kostspielig empfunden und die Teilnahme an Bitkom-Arbeitsgruppen als aufwändig. Was also tun?, fragen sich (und diskutieren) derzeit viele, und zwar sehr kontrovers. Hauptsächlich, aber nicht einziger Grund ist, dass die Bitkom nicht gerade berühmt dafür ist, sich um Open Source zu kümmern.

Und auch dies war und ist Thema, zum Beispiel in einer eigenen BoF-Session auf dem FFG 2013: Wie kann die Weiterentwicklung der GUUG selbst aussehen? Eine „Zukunftswerkstatt GUUG e.V.“ steht auf der mentalen Ideenliste. Angedacht ist, dass der Vorstand hierfür einen Workshop beispielsweise im Linuxhotel organisiert.

Allerdings: Am Vorstand hängt bei weitem nicht alles, und so soll es auch nicht sein. Die SAGE- und regionalen Listen setzen noch nicht einmal eine Vereinsmitgliedschaft voraus. Wer jedoch daran interessiert ist, gemeinsam mit anderen innerhalb der GUUG etwas zu gestalten, für solche gibt es eine Reihe Vereins-Mailinglisten.

Info

„Aus MV und Vorstand“, Uptimes 2012-2, S. 8 - 9: <http://www.guug.de/uptimes/2012-2/index.html>

Über Anika



Anika Kehrer ist freie IT-Journalistin und seit der Wiederauflage der Uptimes im Sommer 2012 als Chefredakteurin an Bord. In dieser Rolle hält sie die Fäden in der Redaktion zusammen und akquiriert und betreut die Artikel der redaktionellen Sommer- und Winterausgaben. Leitidee ist, die Uptimes als Fachmagazin und als Vereinszeitschrift der GUUG anspruchsvoll und transparent zu gestalten, in Zusammenarbeit mit einem Redaktionsteam aus dem Kreis der GUUG-Mitglieder, den restlichen Mitgliedern und dem Vorstand.

Aktiven-Mailinglisten der GUUG

Neben der Vorstandsmailingliste <vs-int@guug.de> existieren die Listen <mitglieder@guug.de> und <guug-members@guug.de>. Die Read-only-Liste „Mitglieder“ ist neu seit Anfang 2012: Sie wurde für die Kommunikation des Vorstandes an die Mitglieder eingeführt. Über diese Liste informiert ein Newsletter quartalsweise (Best-Effort-Prinzip) über Vereinsaktivitäten. Die „GUUG-Members“-Liste ist, im Gegensatz zu der neuen Ankündigungsliste, nicht nur als Leseliste gedacht, sondern der interaktiven Diskussion zwischen Vereinsmitgliedern gewidmet.

Die Uptimes-Chefredaktion kann die GUUG-Mailinglisten übrigens nicht lesen, da dies das Vorrecht der Vereinsmitglieder ist. Für die Mitglieder ergänzt die Uptimes jedoch MV, Vorstandssitzungen und Mailinglisten um die Plattform einer öffentlich publizierten Vereinszeitschrift. Neben dem Herzstück der Uptimes, den Fachbeiträgen über technologische Diskussionen, Methodenbeschreibungen und Einführungen in neue Themen, steht die Uptimes-Rubrik „Vereinsleben“ jedem offen: Ihr Daseinszweck ist, unter Vereins-Internen und -Externen unter den Lesern Impulse zu verteilen OR zum Dialog einzuladen.

In Gedenken an Ulrich Gräf

Unser Kollege und Freund Ulrich Gräf lebt nicht mehr.

von Volker A. Brandt

Uli wurde am 03.12.1959 geboren. Er wuchs in einem kleinen Ort in der Nähe von Gießen auf. Nach dem Abitur 1978 studierte er an der damaligen TH Darmstadt Informatik, wo er auch nach seinem Diplom als wissenschaftlicher Mitarbeiter tätig war.

1992 ging er zu Sun Microsystems, zunächst zuständig für die Finanzindustrie, später dann als Verantwortlicher für das Benchmark-Center in Langen, und weiter als Consultant für verschiedenste Themen im Betriebssystem- und Storage-Bereich. Bekannt wurde er bei vielen Kunden und auf vielen Veranstaltungen als *Solaris Ambassador*, der unermüdlich und bei jeder Gelegenheit die Vorzüge dieses Betriebssystems erläuterte.

Uli war langjähriges GUUG-Mitglied und auch viele Jahre als Vortragender aktiv. Er hat mit einer Ausnahme (2012) bei allen Frühjahrsfachgesprächen von 2005 bis 2013 einen oder mehrere Vorträge sowie Tutorien angeboten. Auch auf den *OpenSolaris Developer Conferences* war er präsent, hat Präsentationen gehalten und weit über den Status eines „Solaris-Lieferanten“ hinaus von diesem System erzählt. So erlebten ihn viele GUUG-Mitglieder und Konferenzbesucher als jemanden, der leidenschaftlich beschreiben und erklären konnte, wie so ein Betriebssystem unter der Haube aussieht und was es im Innersten zusammenhält.

Das war seine hervorstechendste Eigenschaft: die Neugier und Lust an schwierigen Zusammenhängen und komplexen Systemen sowie das Anschaulichmachen dieser Zusammenhänge, sodass auch andere, die nicht die Tiefe seines Wissens besaßen, an seiner Faszination und Entdeckungsfreude teilhaben konnten. Da verwundert es nicht, dass eines seiner Hobbys (neben Unix im allgemeinen und der Datenkompression im besonderen) das Jonglieren war, mit dem er manche Pause oder Wartezeit überbrückt hat.

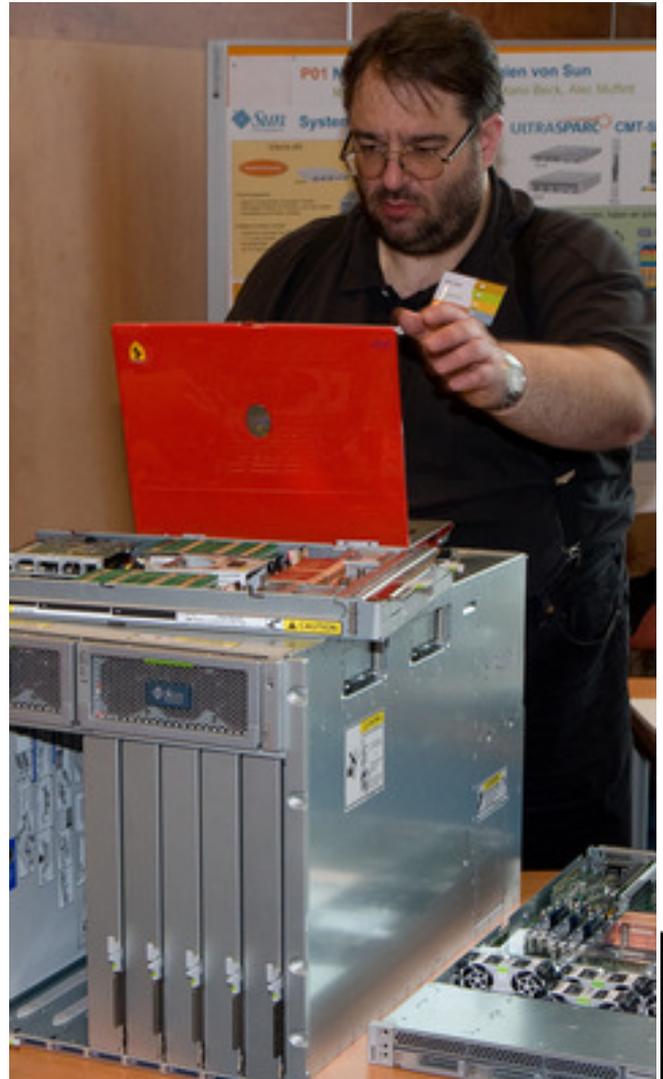


Abbildung 1: Oft gesehen: Mit Ferrari-Laptop auf einem Sun-Blade-Chassis, 2008. (Bild: Franz Haberhauer)

Alle, die ihn kennen, nennen stets dies als ihre wichtigste Erfahrung mit Uli: seine Geduld bei der Darstellung von Wissen, seine Fähigkeit zu erklären, und seine Freigiebigkeit mit seinen Kenntnissen. Er war ein Kommunikator.

Den Open-Source-Gedanken, den Austausch

von Wissen ohne kommerzielle oder institutionelle Schranken, hatte Uli verinnerlicht. Er teilte immer sein Wissen und seine Erfahrung, wenn es nötig war. Viele Sun-Kunden und Kollegen erinnern sich an Begebenheiten, wo er mitgeholfen hat, ein Problem zu analysieren und zu lösen, ohne irgendetwas zurückzuhalten. Dabei war es egal, ob es um Solaris oder Linux oder sonst ein Unix ging. Abzublocken war seine Sache nicht, er wollte den Dingen immer ganz bis auf den Grund gehen. Als Sun den Solaris-Quellcode veröffentlichte und viele Anfragen zu technischen Details bei ihm eintrafen, war er in seinem Element.



Abbildung 2: Beim Vortrag „Solaris ZFS Best Practices“ auf der DOAG (Deutsche Oracle-Anwendergruppe e.V.), 2011. (Bild: Thomas Wagner, der zu der Situation sagt: „Der Vortragstitel untertreibt – es waren wie gewohnt sehr interessante Hintergrundinfos dabei. Uli hatte es eben verstanden und konnte es knackig überbringen.“)

Die andere Richtung des Informationsflusses war ihm ebenfalls nicht fremd; er versuchte für viele Kunden und Solaris-Anwender, Wünsche und Features in Solaris einzubringen, um es für alle besser zu machen. Und oft war er dabei erfolgreich.

Auch an seine Universität, die heutige TU Darmstadt, ist er immer wieder zurückgekehrt und hielt Vorlesungen und andere Veranstaltungen zu Datenbank- und Betriebssystem-Themen. Er erzählte immer begeistert von diesen Gelegenheiten.

Mit der Übernahme von Sun Microsystems durch Oracle war er zu dieser Firma gekommen. Schweren Herzens entschloss er sich Ende 2012, Oracle zu verlassen und sich neuen Aufgaben zu widmen. Ein Buch über Performance-Analyse wollte er schreiben, neue Vorlesungen für die TU Darmstadt zusammenstellen, und vielleicht einen neuen interessanten Job annehmen; mehrere Angebote lagen ihm vor. Im Sommer 2013 sollte es losgehen. Aber seine Pläne kann er leider nicht mehr umsetzen.



Abbildung 3: Beim Jonglieren im Linuxhotel, 2006. (Bild: Johannes Hubertz)

Uli war in jeder Hinsicht groß; wenn er den Raum betrat, war er nicht zu übersehen, und wenn er mit seinem typischen Grinsen eine besonders gute Story erzählte, dann musste man einfach zuhören und in sein lautes Lachen einstimmen.

Er schaute aber auch über den Tellerrand und interessierte sich für viele andere Aspekte der Welt, und wie sie funktioniert. Von Medizin über Astronomie bis Elektrotechnik, über alles konnte man mit ihm reden, und erfuhr immer noch etwas Neues oder Unerwartetes.

Es ist eine besonders bittere und grausame Ironie, dass jemand wie Uli, der so viel „im Kopf“ hatte, so viel konnte und wusste, von einem Hirntumor bezwungen wurde. Am 03.06.2013 starb Ulrich Gräf an den Folgen dieser Krebserkrankung.

Im Namen der GUUG und aller, die Uli kennen, möchten wir Ulis Familie unser tiefes Beileid aussprechen.

Die Unix-Welt ist ohne ihn kleiner und grauer geworden. Wir alle werden ihn sehr vermissen.



Ulrich Gräf bei einem Treffen der Fraosug (Frankfurt Area OpenSolaris User Group), deren Initiator und Motor er war, Januar 2010. (Bild: Michael Gottwald)

PDF-Kung-Foo mit Ghostscript Farbanalyse mit `inkcov`

Dieser Artikel zeigt, wie der Ghostscript-Anwender die in einem PDF verwendeten Farben feststellt und Farb-PDFs, die nur Grauwerte zeigen, in Farbpatronen-schonende Grau-PDFs konvertiert.

von Kurt Pfeifle

In jedem Guru
steckt ein Newbie,
der nie aufgegeben hat.

Ghostscript hat jeder Unix-User schon verwendet, ob bewusst oder unbewusst: zum Beispiel beim Drucken. Ghostscript kann jedoch viel mehr, als nur PostScript-Dateien in ein Format zu wandeln, das zum Zieldrucker passt.

Das aktuellste Ghostscript-Release ist vom Februar und trägt die Versionsnummer 9.07. Seither ist selbst für die Windows-Builds ein neues Device verfügbar: Mit `inkcov` – abgeleitet aus den englischen Begriff „ink coverage“ für Farbabdeckung – kann Ghostscript für jede einzelne Seite anzeigen, welche relativen Tinten- oder Toner-Anteile die Farb-Kartuschen von Cyan, Magenta, Yellow und Black beim Ausdrucken verbrauchen würden.

Damit wird Ghostscript zu einem Tool, um sowohl PDF-, als auch PostScript- oder EPS-Dateien auf relativ unkomplizierte Weise auf verwendete Farben zu prüfen. Das ist übrigens in erster Linie das Verdienst von Sebastian Kapfer von der Universität Erlangen, dessen entsprechenden Feature-Patch die Ghostscript-Entwickler Ende 2011 erstmals in den Quellcode aufnahmen.

Selbstverständlich ist `inkcov` noch nicht sehr breit getestet – viele Anwender wissen ja noch gar nicht, dass es existiert. Wer beim Testen also Fehler oder andere Unzulänglichkeiten findet: Bitte nicht zögern, diese im Ghostscript-Bugtracker zu melden.

Bevor es `inkcov` gab, musste man sich ziemlich verrenken, um ähnliche Informationen zu erhalten. Man musste beispielsweise die zu bewertende Druck-Datei mit Ghostscript seitenweise in eine Serie von Rasterbildern konvertieren (etwa TIFF, PNM, PPM, JPEG oder ähnliches) und diese Bilder dann mittels *ImageMagick* oder Ähnlichem auswerten, um die Anteile der jeweils verwendeten Grundfarben zu bestimmen.

Erstes Beispiel, Input: PDF

Verbrauchsinformation sind für manche Leute von erheblichem Nutzen. Manche Organisationen rechnen die Druckkosten nicht nur pauschal per Seite ab, sondern auch entsprechend ihres Toner- oder Tintenverbrauchs. Oft wäre es auch für sparsame Privatanwender ganz schön, den Verbrauch zu kennen, bevor der Tintenstrahler die teuren Farbpatronen leer pumpt – bloß um daraus große Graufächen zu generieren, die man genauso gut aus reinen Schwarz-Pigmenten hätte komponieren können.

Folgendes Kommando untersucht die Input-Datei `cmyk.pdf`:

```
gs -o - -sDEVICE=inkcov cmyk.pdf
```

Es zeitigt diese Ausgabe:

```
GPL Ghostscript 9.07 (2013-02-14)
Copyright (C) 2012 Artifex Software, Inc.
All rights reserved.
This software comes with NO WARRANTY:
see the file PUBLIC for details.

Page 1
0.00000 0.00000 0.00000 0.02230 CMYK OK
Page 2
0.02360 0.02360 0.02360 0.02360 CMYK OK
Page 3
0.02525 0.02525 0.02525 0.00000 CMYK OK
Page 4
0.00000 0.00000 0.00000 0.01982 CMYK OK
Page 5
0.03306 0.03306 0.03306 0.03355 CMYK OK
```

Die vier Zahlen entsprechen den Werten für die vier Farbkanäle Cyan, Magenta, Yellow und Black (in dieser Reihenfolge). Der Wert „1.0000“ würde 100-prozentige Farbabdeckung der jeweiligen Komponente bedeuten. Seiten, die vier mal „0.00000“ ergeben, sind komplett weiß. Wo nur bei den ersten drei Positionen die Null steht, handelt es sich um reine Schwarz-Weiß-Seiten.

Dementsprechend ist das oben getestete PDF auf Seite 1 ohne Farbe, denn die erste Seite benötigt jeweils 0 Prozent Cyan, Magenta oder Gelb, jedoch 2,23 Prozent Schwarz.

Auf Seite 2 kommen laut `inkcov` jeweils identische Anteile aller vier Farbkomponenten zum Einsatz, nämlich je 2,36 Prozent. Das bedeutet allerdings nicht, dass diese Seite in den menschlichen Sinnesorganen auch tatsächlich Farbwahrnehmungen auslöst: Denn es könnte durchaus sein, dass die gesamte Seite in unterschiedlichen reinen Grau-Tönen gestaltet ist (oder graue Schrift verwendet), wobei alle Graustufen durch die gleichmäßige Mischung aus Cyan-, Magenta- und Yellow-Anteilen zustande kommen.

Ähnliches gilt auch für die Seiten 3 bis 5: Auf jeder dieser Seiten sind die CMY-Werte identisch.

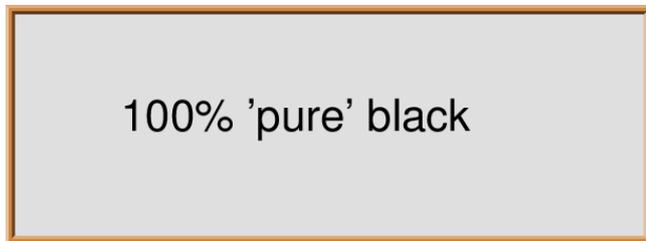


Abbildung 1: Seite 1 von *cmyk.pdf*.

Abbildung 1 visualisiert die erste PDF-Seite als PNG-Bild. Der PostScript-Codeschnipsel, welcher die Farbe der Schrift bestimmt, lautet (weiße Schrift würde `1 setgray` erfordern):

```
0 setgray
```

Auf Seite 2 (Abbildung 2) lautet der PostScript-Codeschnipsel, welcher die Farbe des Textes festlegt:

```
0.5 0.5 0.5 setrgbcolor
```

Das Grau wird hier aus jeweils 50 Prozent der RGB-Farben Rot, Grün und Blau gemischt. Übrigens bezeichnet man diese Technik – nämlich Grau- und Schwarztöne unter Zuhilfenahme von Farbkomponenten zumeist gemeinsam mit Black zusammenzumischen, anstatt dies durch reinen Schwarz-Auftrag zu bewirken – im Grafischen Gewerbe als „Rich Gray“ oder „Rich Black“. Rich Black macht im Druckbild einen satteren Eindruck als ein dünnes pures Schwarz.

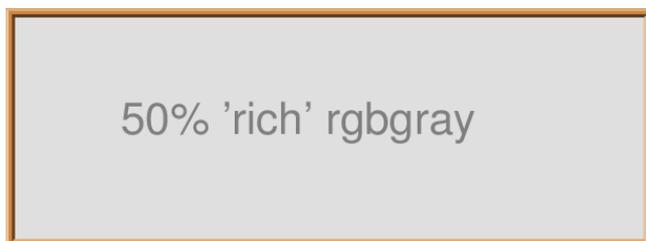


Abbildung 2: Seite 2 von *cmyk.pdf*.

Die dritte Seite (Abbildung 3) verwendet wiederum ausschließlich schwarze Farbe, während die vierte ähnlich wie die zweite zusammengesetzt ist.

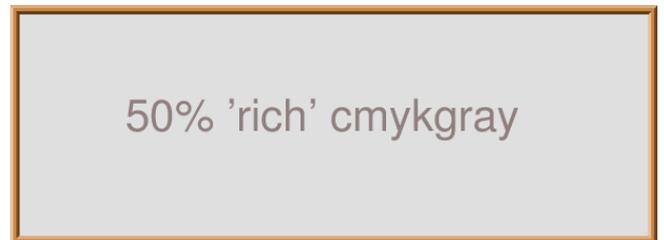


Abbildung 3: Seite 3 von *cmyk.pdf*.

Der PostScript-Codeschnipsel, welcher die Farbgebung des Schriftzuges bestimmt, ist folgender:

```
0.5 0.5 0.5 0 setcmykcolor
```

Das Grau wird hier aus jeweils 50 Prozent der Farben Cyan, Magenta und Yellow gemischt, ohne Zusatz von Black.

Für die vierte Seite der Testdatei (Abbildung 4) setzt folgende PostScript-Anweisung die Schriftfarbe auf ein mittleres Grau:

```
0.5 setgray
```



Abbildung 4: Seite 4 von *cmyk.pdf*.

Die ersten vier Seiten zeigten dem menschlichen Betrachter keinerlei Farben, sondern nur Grautöne an (obwohl deren Quellcode für die Seite 2 den RGB-Farbraum referenziert und für die Seite 3 den CMYK-Farbraum – siehe weiter unten).

Erst auf der fünften Seite (Abbildung 5) sind Farben zu erkennen. Der PostScript-Code, welcher die farbigen Quadrate auf dieser Seite zeichnet, ist etwas komplexer:

```
1 0 0 0 setcmykcolor 100 130 64 64 rectfill
0 1 0 0 setcmykcolor 200 130 64 64 rectfill
0 0 1 0 setcmykcolor 300 130 64 64 rectfill
0 0 0 1 setcmykcolor 400 130 64 64 rectfill
0 1 1 0 setcmykcolor 100 30 64 64 rectfill
1 0 1 0 setcmykcolor 200 30 64 64 rectfill
1 1 0 0 setcmykcolor 300 30 64 64 rectfill
1 1 1 0 setcmykcolor 400 30 64 64 rectfill
```



Erläuterung der vorkommenden Ghostscript-Parameter

<code>-q</code>	Ghostscript soll weniger geschwätzig sein, und zum Beispiel sein „Banner“ nicht nach <i>stderr</i> schreiben.
<code>-o -</code>	Ghostscript soll seine Ausgabe nicht in eine Datei, sondern nach <i>stdout</i> schreiben.
<code>-o cmyk.pdf</code>	Ghostscript soll seine Ausgabe in der Datei <i>cmyk.pdf</i> speichern.
<code>-o inkcov.txt</code>	Ghostscript soll seine Ausgabe in der Datei <i>inkcov.txt</i> speichern.
<code>-g5950x2105</code>	die Seitengröße der Ausgabedatei soll 595 PostScript-Punkte breit sein (wie DIN A 4) und 210,5 PostScript-Punkte hoch.
<code>-sDEVICE=inkcov</code>	Ghostscript soll mit Hilfe des <i>inkcov</i> -Devices die Farbzusammensetzung der Seiten (im CMYK-Farbraum) anzeigen.
<code>-sDEVICE=pdfwrite</code>	Ghostscript soll aus dem Input mittels des <i>pdfwrite</i> -Devices eine PDF-Datei erzeugen.
<code>-c "..."</code>	Ghostscript soll jeden auf <code>-c</code> folgenden String als PostScript-Code interpretieren.
<code>-dFirstPage=323</code>	Ghostscript soll die ersten 322 Seiten der Input-Datei überspringen.
<code>-dLastPage=325</code>	Ghostscript soll ab Seite 326 die Input-Datei nicht mehr weiterverarbeiten.
<code>-sColorConversionStrategy=Gray</code>	Zusammen mit dem nächsten Parameter bewirkt er, dass Ghostscript alle Farben in Grautöne konvertiert.
<code>-sProcessColorModel=DeviceGray</code>	s.o.

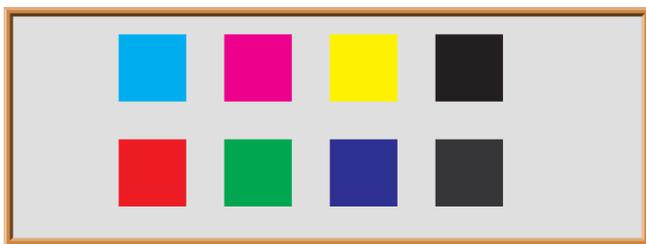


Abbildung 5: Seite 4 von *cmyk.pdf*.

Auf Seite 5 sind Farben deutlich sichtbar – auch wenn ihre Ausgabe von *inkcov*, ebenso wie für die grau-schwarzen Seiten, jeweils identische CMY-Werte anzeigt. Grund ist, dass die Farben (außer beim letzten Quadrat) jeweils rein auftreten, oder aber in Mischungen, welche eben kein Grau ergeben. Lediglich in der Gesamtsumme des Farbauftrags auf der Seite ergeben sich zufälligerweise wieder identische Werte für jeden Kanal.

Zweites Beispiel, Input: PostScript

Ein weiteres Beispiel, das vermutlich jeder Leser sofort nachvollziehen kann, ist die CUPS-Testseite, die auf jedem halbwegs aktuellen Linux-Rechner zu finden sein dürfte. Auf das Ghostscript-Kommando

```
gs \
-o - \
-sDEVICE=inkcov \
/usr/share/cups/data/testprint.ps
```

gibt *inkcov* aus:

```
0.02844 0.02044 0.02439 0.10151 CMYK OK
```

Demnach haben auf dieser Seite 2,844 Prozent aller Pixel eine Cyan-Komponente, 2,044 Prozent verwenden Magenta, 2,439 Prozent verwenden gelben Toner und 1,0151 Prozent schwarzen.

Da die ersten drei Komponenten unterschiedliche Werte haben, muss auf dieser Seite auf jeden Fall Farbe sichtbar sein.

Schaut man sich allerdings den Quellcode der hier untersuchten CUPS-Testseite an, dann fällt auf: Dort kommt eine direkte Verwendung des CMYK-Farbraums gar nicht vor – hierfür müsste sie den PostScript-Operator *setcmykcolor* aufrufen. Man findet hier lediglich *setrgbcolor* – also Farbtöne, die sich aus den Basiskomponenten Rot, Grün und Blau zusammensetzen.

inkcov zeigt keine RGB-Farbwerte an, selbst wenn in der Input-Datei ausschließlich solche vorkommen. Es wandelt alle Farben intern in den CMYK-Farbraum um und zeigt dann die entsprechenden Kennzahlen an.

Eigenes Test-PDF erzeugen

Die hier gezeigten Ergebnisse kann nachvollziehen, wer die Anfangsdatei *cmyk.pdf* auf seinem Rechner hat. Jedoch braucht niemand sein wertvolles Telekom-Limit anzuzapfen, um die Datei aus dem Internet zu holen. Jeder kann sie sich selbst erzeugen – ebenfalls mit Hilfe von Ghostscript. Die Ghostscript-Anweisung dafür lautet:

```
gs \
-o cmyk.pdf \
-sDEVICE=pdfwrite \
-g5950x2105 \
-c "/F1(100 100 moveto/Helvetica findfont 42 scalefont setfont)def" \
-c "F1 (100% 'pure' black) show showpage" \
-c "F1 .5 .5 .5 setrgbcolor (50% 'rich' rgbgray) show showpage" \
-c "F1 .5 .5 .5 0 setcmykcolor (50% 'rich' cmykgray) show showpage" \
-c "F1 .5 setgray (50% 'pure' gray) show showpage" \
-c " 1 0 0 0 setcmykcolor 100 130 64 64 rectfill" \
-c " 0 1 0 0 setcmykcolor 200 130 64 64 rectfill" \
-c " 0 0 1 0 setcmykcolor 300 130 64 64 rectfill" \
-c " 0 0 0 1 setcmykcolor 400 130 64 64 rectfill" \
-c " 0 1 1 0 setcmykcolor 100 30 64 64 rectfill" \
-c " 1 0 1 0 setcmykcolor 200 30 64 64 rectfill" \
-c " 1 1 0 0 setcmykcolor 300 30 64 64 rectfill" \
-c " 1 1 1 0 setcmykcolor 400 30 64 64 rectfill showpage"
```

Transformation in reine Grautöne

Falls man jegliche Farbinformation aus einer PDF entfernen möchte, kann man ebenfalls ein Ghostscript-Kommando verwenden:

```
gs \
-q \
-o grayscale.pdf \
-sDEVICE=pdfwrite \
-sColorConversionStrategy=Gray \
-sProcessColorModel=DeviceGray \
cmyk.pdf
```

Vergleicht man die jeweiligen Seiten vom Original *cmyk.pdf* und seinem Ergebnis *grayscale.pdf* auf dem Bildschirm, erkennt man unter Umständen – je nachdem, wie der Monitor kalibriert ist – bei den ersten vier Seiten keinen Unterschied. Die Konvertierung in reine Grautöne springt erst auf der fünften Seite ins Auge, wo im Original die Farben direkt sichtbar waren: Selbst die meisten von Rot-Grün-Fehlsichtigkeit geplagten Menschen müsste hier zumindest etwas „Buntes“ sehen können.

Übrigens hat Ghostscript bei diesem Kommando eine direkte PDF-nach-PDF-Konvertierung durchgeführt. Leider wissen immer noch viel zu wenige Unix-Geeks, dass dies problemlos möglich ist. Stattdessen gehen sie viel zu häufig den Umweg über PostScript mit Rückwandlung nach PDF – ein Umweg, der fast immer unnötig ist und Qualitäts-Einbußen mit sich bringt.

Die Ausgabe-Datei ist jedenfalls in einen anderen „Farb“-Raum überführt: DeviceGray statt CMYK. Spätestens jetzt müsste jeder Farbdruker seine Farbkartuschen schonen. Dies kann man mittels *inkcov* nachprüfen:

```
gs \
-o inkcov.txt \
-sDEVICE=inkcov \
grayscale.pdf
```

In der Ergebnisdatei *inkcov.txt* sollte jetzt folgendes stehen:

```
0.00000 0.00000 0.00000 0.02230 CMYK OK
0.00000 0.00000 0.00000 0.02360 CMYK OK
0.00000 0.00000 0.00000 0.02525 CMYK OK
0.00000 0.00000 0.00000 0.01982 CMYK OK
0.00000 0.00000 0.00000 0.13274 CMYK OK
```

Auf jeder einzelnen Seiten kommt nur noch der K-Kanal zum Einsatz. Alle Seiten-Elemente, welche beim Rendern C-, M- oder Y-Farben benötigen könnten, sind aus der Datei verschwunden.

Farben jetzt definitiv „unbunt“!

Die konvertierte fünfte Seite der Testdatei sieht aus wie in Abbildung 6. Falls jetzt beim Drucken

immer noch Farbe verbraucht wird, dann trifft den Ersteller des PDF jedenfalls keine Schuld mehr. Dann hat nämlich...

- entweder der Druckertreiber beim Wandeln der PDF-Grautöne in Rasterseiten (falls es ein nicht-PostScript-Gerät ist),
- oder das Gerät selbst (falls es PDF- oder PostScript-Daten direkt verarbeiten kann),
... die Farben neu gemischt.

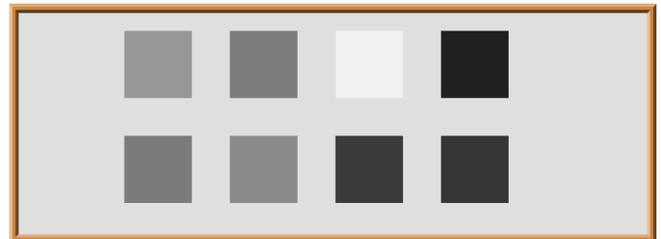


Abbildung 6: Nach der Farbumwandlung zeigt sich Seite 5 jetzt definitiv in grau.

Performance-Tuning

Da Ghostscript zum Berechnen der Werte jede einzelne Seite des Inputs komplett interpretieren und rendern muss, nur um die einzelnen Pixel zu zählen, ist dieser Vorgang praktisch genauso schnell (oder langsam), wie wenn man als Ausgabe eine PNM-Datei (*-sDEVICE=pnm*), PPM-Datei (*-sDEVICE=ppm*) oder JPEG-Datei (*-sDEVICE=jpeg*) herstellen lassen wollte. Ghostscript verwendet für diese Outputs per Default eine Auflösung von 75 dpi (dots per inch). Wer also mittels *inkcov* automatisiert große Mengen von Seiten oder Dateien prüfen und dabei Zeit sparen möchte, reduziert mit Hilfe des zusätzlichen Parameters *-r25* die Auflösung des Berechnungsvorgangs auf zum Beispiel 25 dpi.

Das gröbere Raster der berechneten Bildpunkte bringt natürlich entsprechende Ungenauigkeiten ins Spiel: Die Pixel werden entsprechend größer, nämlich um das jeweils Dreifache in Breite und Höhe, also insgesamt neun mal so groß! Insbesondere feinere Details wie Linien oder Buchstaben in Fließtext erscheinen dadurch fetter, und *inkcov* zeigt folgerichtig einen tendenziell erhöhten relativen Tintenverbrauch an. Man muss selbst austesten, ob die Messungenauigkeit den Performanz-Gewinn zunichte macht.

Wer eine höhere Genauigkeit wünscht, kann natürlich den Parameter in die entgegengesetzte Richtung drehen: Der Wert *-r300* simuliert eine Auflösung von 300 dpi. Der Anwender wartet jedoch entsprechend länger auf das Ergebnis.

Wer aus einer vielseitigen PDF-Datei nur eine einzelne Seite oder eine Range untersuchen möchte, fügt die Kommandozeilen-Parameter `-dFirstPage` und `-dLastPage` hinzu:

```
gs \
-o - \
-q \
-sDEVICE=inkcov \
-dFirstPage=323 \
-dLastPage=325 \
~/Downloads/PDF32000_2008.pdf
```

Dieses Kommando testet nur die Seiten 323 bis 325 der offiziellen PDF-Spezifikation [1] auf ihren Farbauftrag. Folgendes Ergebnis sollte erscheinen:

```
0.00000 0.00000 0.00000 0.05648 CMYK OK
0.00000 0.00000 0.00000 0.03477 CMYK OK
0.00170 0.00170 0.00170 0.05958 CMYK OK
```

Ergebnisse in Datei schreiben

Selbstverständlich kann `inkcov` seine Ergebnisse nicht nur auf die Konsole schreiben (Parameter `-o -`), sondern auch in eine Datei. Hierzu gibt man die gewünschte Zieldatei namentlich an:

```
gs \
-o inkcov-results-PDF32000_2008.log \
-sDEVICE=inkcov \
-dFirstPage=623 \
-dLastPage=655 \
~/Downloads/PDF32000_2008.pdf
```

Dabei wird man feststellen: In obigem Kommando ist der Schalter `-q` nicht vorhanden, damit Ghostscript die Hinweise zur jeweils verarbeiteten Seiten-Nummer nicht verschweigt. Und in der Tat erscheinen auf der Konsole die Meldungen:

```
GPL Ghostscript GIT PRERELEASE 9.08 (2013-01-29)
Copyright (C) 2012 Artifex Software, Inc.
All rights reserved.
This software comes with NO WARRANTY:
see the file PUBLIC for details.
```

```
Processing pages 623 through 655.
Page 623
Page 624
Page 625
[....]
```

In der Logdatei `inkcov-results-PDF32000_2008.log` fehlen diese Angaben jedoch – hier stehen nur die nackten Ergebniszeilen. Offensichtlich gehen die auf der Konsole sichtbaren Meldungen nach `stderr`. Das kann ärgerlich sein, falls man die Logs erst später auswerten möchte, und man dann nicht mehr weiß, welche CMYK-Zeilen zu welcher Seite der untersuchten Datei gehören.

Damit diese Infos ebenfalls in der Logdatei auftauchen, muss man das Kommando wie folgt modifizieren:

```
gs \
-o - \
-sDEVICE=inkcov \
-dFirstPage=623 \
-dLastPage=655 \
~/Downloads/PDF32000_2008.pdf \
> inkcov-results-PDF32000_2008.log
```

Prozentuale Farbkanal-Anteile

`inkcov` kann keine absoluten Werte des Farbverbrauchs angeben, sondern lediglich die prozentualen Anteile einer Farbe bezogen auf die Seitenfläche. Da jeder Farbkanal bis zu 100 Prozent der Fläche bedecken und sich mit den Deckungsbereichen anderer Farben überschneiden kann, kommen bei ganzseitigen Farbflächen auch Summen zusammen, die über 100 Prozent betragen.

Das folgende Ghostscript-Kommando erzeugt eine PDF-Seite, welche diese Umstände demonstriert:

```
gs \
-o one-more-test.pdf \
-sDEVICE=pdfwrite \
-g5950x2105 \
-c "/F1{100 100 moveto/Helvetica findfont 42 scalefont setfont}def" \
-c " 0 0 1 0 setcmykcolor 0 0 595 211 rectfill" \
-c " 1 1 0 0 setcmykcolor 70 20 455 171 rectfill" \
-c " 0 1 1 0 setcmykcolor 140 40 305 131 rectfill" \
-c "F1 1 1 1 1 setcmykcolor (100% 'rich' black) show showpage"
```

Untersucht man das Ergebnis-PDF (Abbildung 7) mit

```
gs -q -o - -sDEVICE=inkcov one-more-test.pdf
```

lautet die Ausgabe:

```
0.31885 0.62139 0.70194 0.02080 CMYK OK
```

Das entspricht einer Gesamt-Farbabdeckung von 166 Prozent mit folgender Aufteilung:

- Cyan 31,885 Prozent
- Magenta 62,139 Prozent
- Yellow 70,194 Prozent
- Black 2,080 Prozent



Abbildung 7: Ergebnis-PDF zum Prozente-Test.

Es ist dem Leser selbst überlassen, herauszufinden, welche Ergebnisse `inkcov` anzeigt, wenn er das vorherige Kommando wie folgt modifiziert, um die Seiten-Dimensionen der Datei zu verdoppeln. Die Werte müssten annähernd gleich sein, da sie als relative Angaben auf die jeweilige Seitengröße bezogen sind. Jeder Drucker wird bei der vierfachen Seitenfläche selbstverständlich die vierfache Farbmenge benötigen:

```
gs
-o one-last-test.pdf \
-sDEVICE=pdfwrite \
-g11900x4210 \
-c "/F1{200 200 moveto/Helvetica findfont 84 scalefont setfont}def\" \
-c " 0 0 1 0 setcmykcolor 0 0 1190 422 rectfill" \
-c " 1 1 0 0 setcmykcolor 140 40 910 342 rectfill" \
-c " 0 1 1 0 setcmykcolor 280 80 610 262 rectfill" \
-c "F1 1 1 1 1 setcmykcolor (100% 'rich' black) show showpage"
```

Foto als PDF-Seite

Untersucht man eine PDF-Seite, welche randlos ein ganzseitiges Digitalfoto anzeigt (wie in Abbildung 8), dann können auch solche Ergebnisse herauskommen:

```
0.96672 0.98699 0.87385 0.59716 CMYK OK
```

Links

[1] PDF-Spezifikation ISO 32000:

http://www.adobe.com/devnet/acrobat/pdfs/PDF32000_2008.pdf

[2] Weitere Tipps und Tricks zu PDF vom Autor:

<http://stackoverflow.com/search?tab=votes&q=user%3a359307%20%5bpdf%5d%20is%3aanswer>

[3] Tipps und Tricks zu PDF-Verarbeitung mit Ghostscript vom Autor:

<http://stackoverflow.com/search?tab=votes&q=user%3a359307%20%5bghostscript%5d%20is%3aanswer>

In der Summe ergibt sich eine (berechnete) Farbabdeckung von über 340 Prozent:

- Cyan 96,672 Prozent
- Magenta 98,699 Prozent
- Yellow 87,385 Prozent
- Black 59,716 Prozent



Abbildung 8: Viel, viel Farbe: Ganzseiten-Foto als PDF mit über 340 Prozent berechneter Farbabdeckung.

Über Kurt



Kurt Pfeifle arbeitet als freiberuflicher IT-Consultant. Er hat sich spezialisiert auf Themen rund um's Drucken und die damit zusammenhängenden Prozesse der Druckdaten-Aufbereitung und -Konvertierung. Am liebsten (aber nicht ausschließlich) setzt er dabei Freie Software ein. Mehr von Kurts Tipps und Tricks zu PDF-Verarbeitung [2] mit Ghostscript [3] kann man zum Beispiel auf StackOverflow finden.

GOOD CODERS...



Wenn der Mini-GAU kommt Kleine Programme zur Sicherung einzelner Rechner

Einzelne Rechner oder kleine Netze lassen sich bequem mit kleinen, einfachen Programmen sichern. Dieser Artikel stellt einige alte und auch neuere Bekannte vor.

von **Stefan Schumacher**

Und überhaupt: so groß zu sein
Ist unmanierlich und nicht fein!

Hermann Löns, Die Zwerge

In der letzten Uptimes 2012-3 gab ein Artikel Tipps für eine Datensicherungsstrategie und ihre Umsetzung [1]. Doch nicht jeder Rechner benötigt eine umfangreiche Datensicherungsstrategie und entsprechend komplexe Programme. Kleine Helfer, die direkt zur Hand sind, stellt daher dieser Artikel vor. Als Grundlage dient das Betriebssystem *NetBSD*, was insbesondere bei *dump* im Hinterkopf zu behalten ist.

dump und restore

dump und *restore* sind historisch gewachsen die bedeutendsten Backup-Programme unter Unix. Sie arbeiten auf Blockebene, also unterhalb der Dateisicht. Das heißt, dass *Dump* einen Verzeichnisbaum oder ein Dateisystem komplett mit allen Eigenschaften sichert, und nicht nur auf Dateiebene Dateien kopiert. Allerdings bleibt *Dump* dabei in der angegebenen Partition, folgt also Mountpoints nicht. Dennoch lassen sich so ganze Partitionen spiegeln und zurücksichern oder auf andere Rechner übertragen.

Da *Dump* an das Dateisystem angepasst ist, kann *dump(8)* nur FFS sichern, für LFS gibt es hingegen *dump_lfs(8)*. Die beiden Programme unterscheiden sich aber nicht in der Handhabung. *Restore* wiederum entpackt einen *Dump* unabhängig vom zu Grunde liegenden Dateisystem, es kann also ein Archiv auch auf einem NFS-Verzeichnis entpacken.

Dump unterstützt inkrementelle Backups durch die Verwendung sogenannter *Dumplevels*. Das Tool trägt bei jedem Backup in der Datei */etc/dumpdates* das entsprechende Datum ein und kann sie so später auslesen und aufbereiten. Ein *Dump* mit *Dumplevel* 0 erzeugt ein Komplettbackup, während ein *Level* größer als 0 die Änderungen seit dem letzten Backup inkrementell

sichert. *Level 2* sichert also alle Daten, die seit dem letzten *Dump* mit *Level 1* hinzukamen oder geändert wurden. In */etc/dumpdates* wandern die Datumsangaben in ein menschenlesbares Format, wenn der Admin die Option *-u* übergibt und *Dump* auf eine ganze Partition ansetzt – nicht aber auf einzelne Dateien oder Verzeichnisse. Mit der Option *-T Datum* lässt sich alternativ ein Wunschdatum angeben, das *Dump* dann als aktuelle Systemzeit betrachtet. Dies ist nützlich für automatisierte Skripte. Die Option *-T Datum* schließt die Option *-u* aus.

Da *Dump* für gewöhnlich dazu dient, komplette Partitionen zu sichern, verfügt es über keine Ein- oder Ausschlusslisten, wie dies einige andere Programme bieten. Stattdessen prüft *Dump*, ob das Fileflag *NODUMP* gesetzt ist. Mit diesem Flag also lassen sich einzelne Dateien oder Verzeichnisse doch vom *Dump* ausschließen. Allerdings ist zu beachten, dass *Dump* mit *Level 0* Fileflags ignoriert. Wer trotzdem die markierten Dateien nicht sichern möchte, übergibt die Option *-h0*.

```
01 $ chflags nodump .signature
02 $ ls -lo .signature
03 -rw-r--r--  1 stefan stefan nodump  \
          369  Feb 18 21:09  .signature
```

Dump lässt sich sowohl auf Dateisysteme als auch auf Partitionen ansetzen. Dies ist ein Anwendungsbeispiel für einen *Dump*-Skript auf mehrere Volumes:

```
01 #!/bin/sh
02 DATUM='date +%y%m%d'
03 QUELLE=/home/
04 # Mediengröße in kB, 102000=>99MB,
05 # 715000=>698MB, 2097152000=>2000MB
06 MEDIUM=102000
07 #Größe des Verzeichnis in kB
08 MEDIEN='du -sk $QUELLE | awk '{print $1}'`
```

```

09 # Medienzahl kalkulieren,
10 # Nachkomma wird abgeschnitten!
11 MEDIEN=$(( $MEDIEN/$MEDIUM ))

12 # Medienzahl aufrunden
13 MEDIEN=$(( $MEDIEN+1 ))

14 # Dateinamen iterieren
15 DAT=${DATUM}_1
16 i=1
17 while [ $i -lt $MEDIEN ]
18 do
19   i=`expr $i + 1`
20   DATEIEN=${DATUM}_$i
21   DAT=${DAT}, ${DATEIEN}
22 done

23 echo "dump -0 -B $MEDIUM -f $DAT $QUELLE"

```

Die Option `-a` in folgenden Kommandos veranlasst `Dump`, die Größe des Archivs selbst festzulegen, anstatt Dateien voreingestellter Größe zu schreiben:

```

01 # dump -0u -f dump-02-31-12 -a /home
02 # dump -0u -f dump-02-31-12 -a /dev/wd0e

```

`Dump` kann mit der Option `-W` auch eine Übersicht der bisher erstellten Dumps aus `/etc/dumpdates` erstellen und mit der Option `-w` die noch zu sichernden Partitionen angeben. Wer Archive bestimmter Größe erzeugen möchte, zum Beispiel für Streamer oder um sie auf eine DVD zu brennen, verfügt mit `dump(8)` über einige, teils historische Optionen. Von Interesse dürfte heute aber nur noch `-B` sein, die als Angabe die Größe der zu sichernden Archive in Kilobyte erwartet. Dazu braucht das Kommando aber zwingend den Namen jeden zu erzeugenden Archivs, da `Dump(8)` sonst automatisch nacheinander in die zuletzt angegebene Datei schreibt. Gibt der Admin beispielsweise drei Dateinamen an, erzeugt aber fünf Dateien, schreibt das Tool die dritte, vierte und fünfte Datei in die dritte. Der unglückliche Admin hätte also drei Sicherungsdateien vorliegen, die aber nur die erste, zweite und fünfte Sicherung enthalten.

`Dump` selbst kann keine Archive komprimieren. Dies besorgt aber eine Pipe an `bzip2` oder `gzip`, oder ein kleines Shellskript. Möglich ist ebenfalls der Einsatz im Netzwerk mittels `rdump(8)` und `rrestore(8)`. Dies ist aber aufgrund mangelnder Sicherheit wie fehlender Verschlüsselung nur in gesicherten Netzwerken ratsam, sodass in anderen Fällen zusätzlich die Umgebungsvariable `RCMD_CMD` einen SSH-Tunnel aufrufen sollte. Noch besser ist eine Pipe an SSH, sodass die Ausgabe auf einem Rechner im Netzwerk landet:

```

01 # dump -0a -f - home | \
    ssh operator@192.168.1.1 \
    dd of=/usr/backup/dump.0

```

Dump im Detail

Da `Dump` das wichtigste Sicherungsprogramm ist, verdient es einen genaueren Blick. Dies ist ein exemplarischer Log eines Sicherungslaufes:

```

01 root@# dump -lau -h0 -f - /home |gzip > home.1
02 DUMP: Found /dev/rwd1a on /home in /etc/fstab
03 DUMP: Date of this level 1 dump: Sat Dec 24 12:17:56 2005
04 DUMP: Date of last level 0 dump: Sun Dec 18 23:20:19 2005
05 DUMP: Dumping /dev/rwd1a (/home) to standard output
06 DUMP: Label: none
07 DUMP: mapping (Pass I) [regular files]
08 DUMP: mapping (Pass II) [directories]
09 DUMP: mapping (Pass II) [directories]
10 DUMP: estimated 41438 tape blocks.
11 DUMP: Volume 1 started at: Sat Dec 24 12:18:18 2005
12 DUMP: dumping (Pass III) [directories]
13 DUMP: dumping (Pass IV) [regular files]
14 DUMP: 40541 tape blocks
15 DUMP: Volume 1 completed at: Sat Dec 24 12:18:40 2005
16 DUMP: Volume 1 took 0:00:22
17 DUMP: Volume 1 transfer rate: 1842 KB/s
18 DUMP: Date of this level 1 dump: Sat Dec 24 12:17:56 2005
19 DUMP: Date this dump completed: Sat Dec 24 12:18:40 2005
20 DUMP: Average transfer rate: 1842 KB/s
21 DUMP: level 1 dump on Sat Dec 24 12:17:56 2005
22 DUMP: DUMP IS DONE
23 root@#

```

Dieses exemplarische Log eines Sicherungslaufes gibt wertvolle Informationen wieder. Das zu sichernde Verzeichnis `/home` ist als eigene Slice `/dev/rwd1a` in `/etc/fstab` aufgeführt (Zeilen 01 und 02). Dieser Sicherungslauf (Level 1) beginnt um 12:17 Uhr am 24.12.2005 (Zeile 03 und Zeile 21). Der letzte Sicherungslauf im Level 0 fand am 18.12. statt (Zeile 04). Der Dump wird durchgeführt (Zeile 05) und erfolgreich beendet (Zeile 22). Dazu wird eine Statistik unter anderem über Dauer, Platz und Übertragungsrates ausgegeben (Zeilen 06 bis 20).

Interessant sind die einzelnen Durchläufe *Pass I* bis *Pass IV* (Zeilen 07 bis 13):

- Aus dem aktuellen Datum, dem angegebenen Level (hier: 1) und den letzten relevanten Durchläufen – aufgezeichnet in `/etc/dumpdates` – berechnet `Dump` die Variable `DUMP_SINCE`. Es traversiert alle Inodes im Dateisystem und vergleicht deren *modification time* (`mtime`) mit der Größe von `DUMP_SINCE`. Ist die *mtime* einer Datei gleich groß oder größer, wandert der Inode in die Liste zu sichernder Dateien. Nicht zugeordnete Inodes ignoriert `Dump`, sodass am Ende des ersten Durchlaufes drei Listen vorliegen: die Inodes der Dateien, die zu sichern sind; die Inodes aller Verzeichnisse; und die zugeordneten Inodes.
- Durchlauf II läuft in mehreren Stufen ab. Als erstes traversiert `Dump` erneut die in Durchlauf 1 erstellte Liste der Verzeichnisse und prüft, ob zu sichernde Dateien im Verzeichnis existieren. Wenn nicht, schmeißt es den entsprechenden Inode aus der Liste der zu sichernden

Verzeichnisse wieder hinaus. Als zweites überprüft es die Verzeichnisse erneut, indem es den Verzeichnisbaum von den Blättern zur Wurzel durchläuft. Dabei findet es im letzten Durchlauf frei gewordene Verzeichnisse. Im Beispiel fand es keine frei gewordenen Verzeichnisse, da sonst ein weiterer *Pass II* stattgefunden hätte.

- Vorbereitungen zu *Pass III*: Bevor Dump mit dem Schreiben der Daten beginnt, legt es deren Metainformationen ab. Hierzu schreibt es einen *Header*, der die nachfolgenden Daten beschreibt. Zusätzlich schreibt das Tool zwei Inode-Tabellen – eine im nativen Format, und eine aus Kompatibilitätsgründen im normalisierten alten BSD-Format.
- Jetzt entsteht ein Header, der den Inode des Verzeichnisses enthält, und die Datenblöcke für jedes Verzeichnis in der Liste der zu sichernden Verzeichnisse werden geschrieben (Zeile 12).
- Jetzt entsteht ein Header, der den Inode der Datei enthält, und die Datenblöcke für jede Datei in der zu-sichernde-Dateien-Liste werden geschrieben (Zeile 13).
- Nachbereitung von *Pass IV*: Der abschließende Header wird aufs Band geschrieben.

Inkonsistenzen in der Sicherung treten auf, wenn das Quelllaufwerk während der Sicherung beschrieben wird. In diesem Fall würden einzelne Dateien nicht oder falsch gesichert. Eine Korruption der gesamten Sicherung ist aber unwahrscheinlich. Vermeiden lassen sich diese Inkonsistenzen, wenn der Backup-Admin das Dateisystem unmountet, oder wenn er Snapshots verwendet.

restore

Restore spielt mittels der Option `-r` ein komplettes Archiv zurück. Dabei erzeugt es eine Datei namens *restoresymtable*, in der es Meta-Informationen zur Rücksicherung ablegt – also Inodes, oder welche Bänder. Diese Datei kann der Admin nach der Rücksicherung aller (!) Archive entfernen:

```
01 # restore -r -f 02-11-01 && \
    restore -r -f 02-11-08
02 # rm restoresymtable
```

Wer nur bestimmte Pfade oder Dateien zurückspielen möchte, übergibt Teile des Verzeichnisses mit der Option `-x`. Leider unterstützt Restore keine regulären Ausdrücke, aber man kann sich mit ein paar Backticks behelfen. Um zu prüfen ob ein Archiv erfolgreich geschrieben wurde, reicht es zu Testzwecken, die allerletzte Datei zu restaurieren.

Da Dump alle vorigen Einträge zumindest traversieren muss, erkennt es Fehler im Archiv und meldet sie. In der Dateiliste (Option `-t`) stehen die Inodes und die korrespondierende Dateinamen, man kann also den letzten Eintrag zurücksichern:

```
01 $ restore -x stefan/ www/
02 $ restore -t > dateiliste
03 $ restore -x `grep -i 'bz2$' dateiliste | \
    awk '{print $2}'`
04 $ restore -x `tail -n 1 dateiliste | \
    awk '{print $2}'`
```

Ist eine bestimmte Datei im gesicherten Verzeichnisbaum gewünscht, ruft `restore -i` eine Shell auf, in der sich mit `ls` und `cd` der Verzeichnisbaum durchkämmen lässt. Ist die gewünschte Datei gefunden, fügt `add` sie zur Liste der zurückzusichernden Dateien hinzu. Mit `extract` werden wiederum die in der Liste spezifizierten Dateien im aktuellen Verzeichnis entpackt. Um einen unterbrochenen Lauf fortzusetzen, dazu dient die Option `-R`.

tar(1)

tar(1) ist das wohl bekannteste Sicherungsprogramm. Allerdings gibt es verschiedene Versionen und Implementierungen, sodass man darauf achten sollte, nur kompatible Versionen einzusetzen und mithin kompatible Archive zu erzeugen.

Tar steht für „Tape Archiver“, es entstand also zu dem Zweck, Bandarchive zu beschreiben. Es erzeugt dazu eine Datei, die alle zu sichernden Dateien enthält, und schreibt diese auf ein Band – oder in eine einfache Datei. Die Option `f` erzeugt eine Datei – standardmäßig würde Tar das Archiv sonst auf das erste Bandlaufwerk `/dev/nrst0` schreiben.

```
01 $ tar cpfz \
    operator@backup:/home/backup/backup.tgz \
    /home/
02 $ tar xpfz backup.tgz
```

Tar schreibt relativ zum jetzigen Pfad, das im Beispiel gesicherte Home-Verzeichnis wird also an aktueller Stelle entpackt. Die Option `z` erzeugt ein Gzip-Archiv. Mit `-j` ließe sich auch ein Bzip2-komprimiertes Archiv erzeugen. Zurückspielen lässt sich das Backup mit `x`. Die Option `p` sorgt dafür, dass die Dateirechte bestehen bleiben:

Es ist wichtig zu wissen, dass Tar große Probleme mit defekten Archiven hat, da bei einem Archiv ein Header erzeugt wird, der Informationen über das gesamte Archiv enthält. Taucht an einer Stelle ein Fehler auf, entpackt Tar das Archiv nur bis zu dieser Fehlerstelle.

star

star ist eine von Jörg Schilling seit 1982 entwickelte Tar-Implementierung. Es ist die weltweit schnellste Tar-Variante und besitzt einige nützliche Erweiterungen. Es unterstützt unter anderem reguläre Ausdrücke, Levels (so wie Dump), begrenzt die Pfadlänge nicht, erkennt automatisch die Endianness und ermöglicht, ein Dateisystem mit einem Tarball zu vergleichen. In der Standardvariante kann Star Tar-Archive lesen und umgekehrt.

Star lässt sich aus *pkgsrc/archivers/star* installieren und ist im Allgemeinen dem normalen Tar der Vorzug zu geben. Star unterstützt verschiedene Tar-Formate, die mit der Option `-H=` bestimmt werden können. Am leistungsfähigsten ist *EXU-STAR*.

cpio(1)

cpio(1) (von „copy in/out“) ist von der Syntax her etwas eigen, da es als erste Option die Übergabe einer zu sichernden Datei erwartet und diese auf *STDOUT* schreibt. Das erweist sich allerdings mit Hilfe der Pipe als Vorteil, da der Kommandeur so mit *find(1)* oder *ls(1)* die Dateiliste übergeben und die Ausgabe mit `>` umleiten kann. Dateien mit Cpio zu sichern, sieht folgendermaßen aus:

```
01 $ ls *.jpg | cpio -0 > /home/Bilder.cpio
02 $ find /home -mtime 7 | cpio -o > /dev/nrst0
03 $ cpio -i < /dev/nrst0
```

Dieses Listing zeigt in Zeile 01, wie man alle JPGs im aktuellen Verzeichnis in die Datei *Bilder.cpio* sichert. Die Ausgabe kann stattdessen auch auf das Bandlaufwerk */dev/nrst0* umleiten. Um Unterverzeichnisse zu sichern oder inkrementelle

Backups zu erzeugen, nutzt man *find(1)*. Zeile 02 sichert alle Dateien, die in den letzten sieben Tagen geändert wurden, auf das erste Bandlaufwerk. Zeile 03 spielt schließlich die Sicherung zurück.

Leider unterstützt Cpio keine Überprüfung der Archive, sodass dies manuell zu erledigen ist. Dazu entpackt man das Archiv wieder und vergleicht die Dateien mit *diff* oder *mtree*.

afio(1)

afio(1) ist eine Neuauflage von Cpio und behebt einige Probleme, etwa mit leeren Dateien, oder mit 0 gesetzten Rechten. Es verwendet dieselbe Syntax wie Cpio. Das Tool ist via *pkgsrc/archivers/afio* installierbar und sollte Cpio vorgezogen werden, ebenso wie Star bei Tar.

pax(1)

pax(1) ist ein Standardisierungsversuch der IETF, da es zu viele Implementierungen von Tar und Cpio gibt. Pax ist in der Lage, verschiedene Archivformate wie Cpio und Tar zu schreiben oder zu lesen. Daher ist es in heterogenen Umgebungen nützlich. Das Tool wird unter anderem auch auf den Installationsmedien von NetBSD verwendet. So sieht eine Sicherung und Rücksicherung mit Pax aus:

```
01 # pax -w -f /dev/nrst0 /home
02 # pax -v -f /dev/nrst0
03 $ pax -r -s ',^/*usr/* »' -f a.pax
```

Zeile 01 schreibt das Home-Verzeichnis auf Band. Zeile 02 gibt ein Inhaltsverzeichnis der Sicherung aus. Der letzte Befehl packt alle Dateien in */usr* in das Archiv *.a.pax*.

Literatur

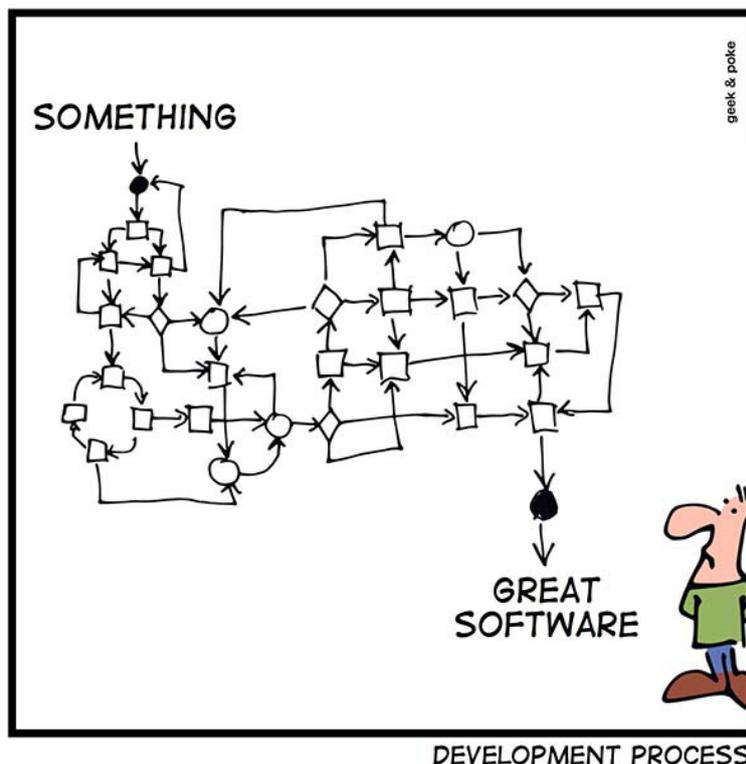
[1] Stefan Schumacher: Wenn der GAU kommt. Datensicherungsstrategie erarbeiten und umsetzen, Uptimes 2012-3, S. 21-29: <http://www.guug.de/uptimes/2012-3/index.html>

Über Stefan



Stefan Schumacher ist geschäftsführender Direktor des Magdeburger Instituts für Sicherheitsforschung und gibt zusammen mit Jan W. Meine das Magdeburger Journal zur Sicherheitsforschung heraus. Er befasst sich seit knapp 20 Jahren als Hacker mit Fragen der Informations- und Unternehmenssicherheit und erforscht Sicherheitsfragen aus pädagogisch-psychologischer Sicht. Seine Forschungsergebnisse stellt er auf Fachkongressen und in Publikationen vor. Seine Schwerpunkte liegen auf Social Engineering, Security Awareness, Organisationssicherheit, internationale Cyber-Security und Mensch-Maschine-Interaktion.

SIMPLY EXPLAINED



Shellskripte mit Aha-Effekt Erlaubte Zeichen in E-Mail-Adressen

Alte Hasen wissen, dass mancher RFC Geheimnisse birgt, weshalb ihre Lektüre durchaus Kurzweil und Schabernack ermöglicht – insbesondere bei RFCs, deren Erstellungsdatum zufällig auf den ersten April eines Jahres fällt. Der Autor hat sich angesehen, was man mit E-Mail-Adressen alles machen kann.

von Jürgen Plate

Läuten Feierabendglocken
Hacken Admins unerschrocken
Auf die Tastatur geschwind,
Schelmisch sinnend wie ein Kind.

Frei nach Joseph von Eichendorff, Der Schalk

Das Format eine E-Mail-Adresse ist in RFC 2822 festgelegt, der nur eine Untermenge des ASCII-Zeichensatzes in E-Mail-Adressen erlaubt. In RFC 1642 wird jedoch UTF-7 definiert und damit eine Möglichkeit geboten, alle Unicode-Zeichen unter Verwendung des beschränkten ASCII-Zeichensatzes darzustellen. E-Mail-Adressen, die das ausnutzen, sind fast nur in asiatischen Ländern in Gebrauch.

Bekanntlich besteht eine E-Mail-Adresse aus dem *local-part*, der vor dem Klammeraffen steht, und dem *domain-part*, der dahinter steht. Wie in RFC 2821 festgelegt, darf der Local-part einer E-Mail-Adresse bis zu 64 Zeichen lang sein, und der Domain-part bis zu 253 Zeichen.

Nach RFC 2822 darf der Local-part außerdem folgende Zeichen enthalten:

- Die Groß- und Kleinbuchstaben (a bis z, A bis Z) (ASCII: 65-90, 97-122). Obwohl der RFC zwischen Groß- und Kleinschreibung unterscheidet, ignorieren dies die meisten Mail-systeme und konvertieren die Großbuchstaben in Kleinbuchstaben.
- Die Ziffern 0 bis 9 (ASCII: 48-57).
- Die Sonderzeichen ! # \$ % & ' * + - / = ? ^ _ ` { | } ~ (ASCII: 33, 35-39, 42, 43, 45, 47, 61, 63, 94-96, 123-126).
- Den Punkt (.) (ASCII: 46), sofern er nicht an erster oder letzter Stelle steht und er nicht mehrmals aufeinander folgt (zum Beispiel ist *Sepp..Huber@beispiel.de* verboten).

Eingeschränkt ist auch noch Folgendes erlaubt:

- Leerzeichen (ASCII: 32).
- Die Sonderzeichen " () , : ; < > @ [\] (ASCII: 34, 40, 41, 44, 58, 59, 60, 62, 64, 91-93). Diese Zeichen sind nur erlaubt, wenn sie in Anführungszeichen eingeschlossen werden.

- Leerzeichen, Backslash und Anführungszeichen (ASCII: 32, 92, 34) müssen außerdem durch einen Backslash maskiert werden.
- Zusätzlich sind Kommentare in Klammern erlaubt, beispielsweise sind die drei E-Mail-Adressen *sepp.huber(der Bazi)@dings.com*, *sepp(der Bazi).huber@dings.com* und *jo(se)f.huber@dings.com* gleichbedeutend mit *sepp.huber@dings.com*.
- Internationale Zeichen oberhalb von U+007F sind durch RFC 6531 erlaubt, werden aber nicht von allen Mail-Systemen akzeptiert.

Ein String in Anführungszeichen muss entweder ein durch Punkt abgesonderter Teil des Local-part sein, oder es muss der gesamte Local-part in Anführungszeichen stehen. Zum Beispiel ist erlaubt:

```
abc."defghi".xyz@example.com
"abcdefghixyz"@example.com
```

Verboten ist jedoch:

```
abc"defghi"xyz@example.com
abc\"def\"ghi@example.com
```

Strings in Anführungszeichen werden jedoch fast nie verwendet. RFC 5321 warnt sogar davor: "A host that expects to receive mail SHOULD avoid defining mailboxes where the Local-part requires (or uses) the Quoted-string form."

Der Domain-part wird dagegen restriktiver behandelt. Er ist beschränkt auf Buchstaben, Ziffern, den Punkt und das Minuszeichen. Hier werden auch Groß- und Kleinschreibung nicht unterschieden. Anstelle eines Domain-Namens darf auch eine IP-Adresse in eckigen Klammern stehen – etwa *jim.beam@[192.168.222.111]* – obwohl diese Form der Adressierung fast nur Spammer verwenden. Auch im Domain-part sind geklammerte Kommentare erlaubt, wie etwa *johnny.walker@(comment)example.com*.

Wer jetzt mit seltsamen Mailadressen spielen möchte, findet erstens abschließend noch ein paar Beispiele. Zweitens denkt sie oder er daran, dass nicht jedes Mailsystem das auch akzeptiert – selbst wenn es im RFC steht, verhindern kluge Admi-

nistratoren den geschilderten „Wildwuchs“. Was klappt, gilt es auszuprobieren. Immerhin ist der Schreiber dieser Zeilen auch unter ****@netzmafia.de* zu erreichen.



Beispiele für gültige E-Mail-Adressen

```
einfachundgeschmackvoll@beispiel.de
einfach+geschmackvoll@beispiel.de
schon.anders@beispiel.de
ein.wenig.komplizierter@dept.beispiel.de
nobody@[IPv6:2001:db8:1ff::a0b:dbd0]
"etwas.seltsam\ ist\ dies"@beispiel.de
"sehr.seltsam.@unusual.org"@beispiel.de
"sehr.(), ;;<>[]\".SEHR.\"sehr@\\\"sehr\".seltsam"@beispiel.de
0@a
!#$%&'*+,-/=/?^_`{|}@netzmafia.de
"@beispiel.de
```



Beispiele für ungültige E-Mail-Adressen

```
Abc.beispiel.de (Denn: Der Klammeraffe muss Local-part und Domain-part trennen.)
Abc.@beispiel.de (Denn: Der Punkt ist letztes Zeichen des Local-part.)
Abc..123@beispiel.de (Denn: Der Punkt kommt zweimal hintereinander vor.)
A@b@c@beispiel.de (Denn: Nur ein Klammeraffe außerhalb der Anführungszeichen ist erlaubt.)
a"b(c)d,e:f;g&lt;h&gt;i[j\k]l@beispiel.de (Denn: Diese Sonderzeichen müssen innerhalb von Anführungszeichen stehen.)
das"ist"falsch@beispiel.de (Denn: Strings in Anführungszeichen müssen durch Punkte abgetrennt werden.)
das ist"nicht\erlaubt@beispiel.de (Denn: Leerzeichen, Anführungszeichen und Backslashes müssen durch Backslash maskiert werden.)
auch\ das\"ist\\nicht\ erlaubt@beispiel.de (Denn: Auch wenn die Zeichen maskiert sind, sind Anführungszeichen um den ganzen Local-part notwendig.)
```

Geschichtsstunde III

1990 - heute: Unix hält die Fahne hoch

Fast hätte es Unix erwischt, als Novell die Rechte an der Marke erwarb und dann doch kein so rechtes Interesse daran hatte. Die Bemühungen von SCO Group, auf dem Klageweg ein Almosen zu ergattern, waren nicht geeignet, die Lust zu steigern. Aber es kam dann gerade umgekehrt. Schuld war ein Student aus Helsinki.

von Jürgen Plate

Der letzte Teil der dreiteiligen Geschichtsstunde [1, 2] steigt in den 1990er Jahren ein, als die *Unix System Laboratories* (USL) 1991 gemeinsam mit *Novell* das Jointventure *Univel* gründeten, bei dem *Novell* die Majorität hatte. Sie veröffentlichten weitere Unix-Versionen unter dem Namen *UnixWare*. Im Jahr 1995 wurde auch das Betriebssystem *SCO UNIX* zu *SCO OpenServer* umbenannt. Die danach geplante Verschmelzung mit *UnixWare* fand 2005 im aktuellen Release 6 statt, welches auf dem *Unix-SVR5*-Systemkern aufbaut. In Deutschland gab es sogar eine kurze Anleitung dazu, die von Tiki Küstenmacher gezeichnet wurde. Küstenmacher hat damals übrigens auch recht erfolgreiche Comichbücher zu *MS-DOS*, *Word* und anderem veröffentlicht. Die *Open Software Foundation* (OSF) stellte unterdessen die erste Version ihres Betriebssystems *OSF/1* als Konkurrenz zu *UNIX System V*, *UnixWare* und anderen Unix-Systemen vor.

Novell erwarb 1993 auch die übrigen Anteile an der *Unix*-Entwicklungsabteilung sowie sämtliche Rechte am System. 1995 wurde die Entwicklungsabteilung an die *SCO-Group* weiterverkauft – das ist die, über die ich in Teil II schrieb, dass sie nichts mit der honorigen, aber insolvent gewordenen Firma *SCO* zu tun hat. Das nahm *SCO* zum Anlass, um im Jahr 2003 mehrere Unternehmen wegen Urheberrechtsverletzungen zu verklagen. Der Prozess endete 2009 mit dem Ergebnis, dass *Novell* nach wie vor im Besitz der Rechte an *UNIX* ist: Die Anschuldigungen von *SCO* erwiesen sich durchweg als haltlos. Die Idee, auf dem Klageweg zu fetten Lizenzeeinnahmen zu kommen, löste sich in Wohlgefallen auf, und *SCO* wurde wegen der zehrenden Gerichtsverfahren zahlungsunfähig.

Novell signalisierte durch den Verkauf der Entwicklungsabteilung, dass an der Weiterentwicklung von *Unix* kein Interesse bestand. Ist damit

Linux is a cancer that attaches itself in an intellectual property sense to everything it touches.

Steve Ballmer, Juni 2001

Will we interoperate with products that come, like Linux, from the open source world? Yes, we will. Will we encourage people who want to do open source development to do it on top of Windows? Yes.

Steve Ballmer, Juli 2008

das Ende der Entwicklung von *Unix* besiegelt? Die Antwort lautet natürlich: Nein. Mit den *BSD*-Varianten *OpenBSD*, *FreeBSD* und *NetBSD* standen bereits drei *Unix*-Vertreter zur Verfügung. Und auch *Novell* kam nicht ganz los von *Unix*, denn durch den späteren Kauf der Firma *S.u.S.E* kam es durch die Hintertür wieder herein.

Die BSD-Familien

OpenBSD (Abbildung 1) ist unter der sogenannten *BSD-Lizenz* frei verfügbar. Es wurde bereits 1994 durch Theo de Raadt von *NetBSD* abgespalten. *OpenBSD* ist bekannt für das Beharren seiner Entwickler auf Quelloffenheit, freier Dokumentation (auch wenn die Manualpages manchmal sehr knapp gehalten sind) und kompromissloser Haltung gegenüber Software-Lizenzen. Insbesondere handelt es sich um ein auf Computersicherheit und Stabilität hin gezüchtetes *Unix*.



Abbildung 1: Das Maskottchen von *OpenBSD*: Puffy, der Kugelfisch.

Auch *NetBSD*, das ja nach *386BSD* die zweite Quelloffene *BSD*-Variante war, wird bis heute aktiv weiterentwickelt und ist zudem gut auf andere

Plattformen portierbar. Mittlerweile läuft NetBSD auf 57 Hardware-Plattformen. Auch bei NetBSD werden Qualität und klarer Aufbau allseits gelobt. Nur wenig später wurde auch FreeBSD gegründet. Auch hier wird das System bis heute aktuell gehalten.

FreeBSD war als Server-Unix konzipiert und wird daher von vielen Internet Providern, in Internet-Backbone-Systemen und bei Webhosting-Betreibern eingesetzt. Auch Apples Betriebssystem *Mac OS X* (Darwin) basiert auf FreeBSD 3.0 und dem Mach-Mikrokern. Der Sourcecode ist frei verfügbar und genauso portabel wie FreeBSD. Apple sponsort das Projekt und gibt Verbesserungen an das FreeBSD-Projekt zurück.

Windows bekommt gerade noch den Anschluss

Während sich in den 1990er Jahren die kommerziellen Unix-Entwickler um Marktanteile balgten, brachte Microsoft 1995 *Windows 95* und ein Jahr später *Windows NT 4.0* heraus – 32-bit-Windows-Versionen mit preemptivem Multitasking und grafischer Oberfläche. Beim Launch von *Windows 95* war zwar noch ausschließlich von Mailboxsystemen wie *CompuServe*, *AOL* oder dem im Aufbau befindlichen *Microsoft Network* (MSN) die Rede. Doch gelang es Microsoft gerade noch, mit Hilfe von BSD auf den abfahrenden Internet-Zug aufzuspringen. Das ist noch heute an Kommandozeilentools wie *ping*, *tracert* oder *ftp* erkennbar, die genau wie ihre BSD-Vettern aussehen.

Mit *Windows NT* und später *Windows 2000* zielte Microsoft ganz klar auf die Unix-Domänen, also Workstations, Webserver oder Fileserver. Es zeigte sich jedoch, dass sie mitsamt ihrer Nachfolger Mächtgern-Unixe sind: Die als innovativ bezeichneten Konzepte wie Hardlinks oder Mounten von Partitionen und Platten gibt es unter Unix seit mehr als 25 Jahren.

Was machen die bedauernswerten Adepten der reinen Lehre, wenn sie in ihrer Firma nur *Windows-PCs* vor die Nase gesetzt bekommen? Die Frickel-Textverarbeitung kann man ja gerade noch mit Hilfe einer \LaTeX -Installation umgehen, aber die kleinen Helfer *vi*, *emacs*, *find*, *grep*, *awk*, *tar* und natürlich die Shell fehlen einfach.

Cygwin – noch ein Exot

Zum Glück begann 1995 Steve Chamberlain, Software-Entwickler bei *Cygnus Solutions*, den GNU C-Compiler *GCC* auch unter dem neuen

Windows 95 und *NT* lauffähig zu machen. Da die GNU-Compiler bereits Unterstützung für x86 und COFF in Zusammenhang mit der C-Bibliothek *newlib* boten, war recht bald eine Cross-Plattform geschaffen, die Code für *Windows* erzeugen konnte.

Um den Compiler zu portieren, wurden die ganzen Hilfsprogramme benötigt, darunter *make* und *bash*. Die *Win32-API* enthielt bereits etliche sehr ähnliche Funktionen wie Unix. So waren lediglich die meisten Systemaufrufe anzupassen. Dies führte dann zu den *Cygwin-DLLs*, die direkt auf dem *Windows*-System liefen, aber nach oben hin Unix-Dienste leisteten. In den folgenden Jahre wuchs die Zahl der portierten Tools. 1998 begann Cygnus, *Cygwin* gewinnbringend zu vermarkten.

Mit *Cygwin* lassen sich Programme, die üblicherweise unter Posix-Systemen laufen, auf Microsoft *Windows* portieren. Mit *Cygwin* portierte Programme laufen unter *Windows NT*, *Windows 2000*, *Windows XP*, *Windows Vista*, *Windows Server 2003* und seit Version 1.7 auch unter *Windows 7* und *Windows Server 2008*. Mit *Cygwin/X* existiert auch eine Portierung des X.Org-Servers für die *Cygwin*-Umgebung, so dass unter *Windows* ein kompletter X-Server bereitsteht. Lokale Partitionen werden über */cygdrive/c*, */cygdrive/d* und so weiter angesprochen. Ebenso kann auf *Windows*-Freigaben zugegriffen werden.

Heute steht *Cygwin* als Open-Source-Software unter der *GPL*. Ende 1999 übernahm *Red Hat* die *Cygnus Solutions* und ist seitdem auch für *Cygwin* verantwortlich.

Sun und Solaris

Die Geschichte wäre nicht vollständig, ohne das 1982 gegründete Unternehmen *SUN Microsystems*. Die erste Baureihe der Sun-Workstation lief mit *SunOS*, einem BSD-4.3-Klon. 1984 entwickelte Sun mit *NFS* (Network File System, oder auch: „Nightmare File System“) ein Netzwerkdateisystem, das unter einer Open-Source-Lizenz stand und so in der Unix-Welt bis heute weit verbreitet ist. Mit dem *SPARC*-Prozessor (Bild 2) hatte Sun eine sehr leistungsfähige RISC-CPU für die eigenen Workstations und Server an Bord, auf der erst *SunOS* und später *Solaris* lief.

Mit *Java* schuf Sun 1995 eine objektorientierte, systemunabhängige Programmierplattform, die schnell viele Anhänger fand. 1999 kaufte Sun das Unternehmen *Star Division* mit dem Office-Paket

StarOffice, das ab 2000 unter der freien Lizenz *LG-PL* als *OpenOffice* veröffentlicht wurde und sich schnell verbreitete. Anfang 2008 übernahm Sun auch das Unternehmen *MySQL AB*, das die Open-Source-Datenbank *MySQL* entwickelt und vermarktet hat. Denselben Weg ging kurz darauf *Innotek*, Hersteller der Virtualisierungslösung *VirtualBox*.

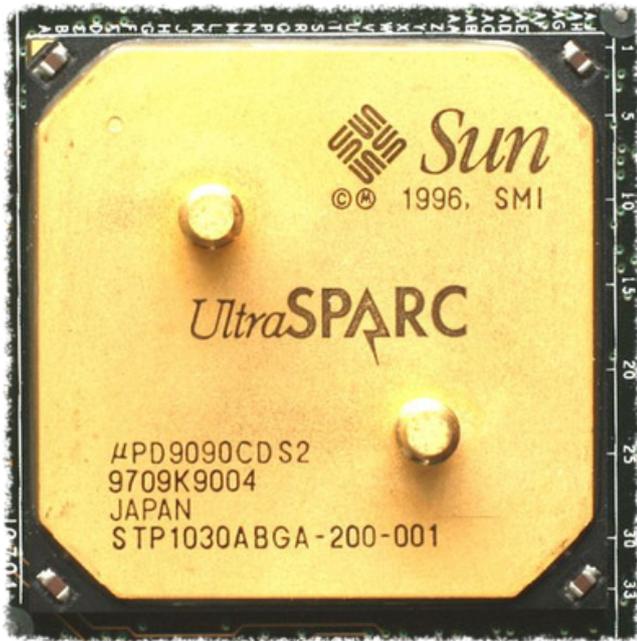


Abbildung 2: Ultra-Sparc-Prozessor von Sun. Gut erkennbar sind die beiden Bolzen zum Anschrauben des Kühlkörpers. (Bild: Konstantin Lanzet)

Alles schien gut. Jedoch musste Sun 2009 einen Verlust von 2,2 Milliarden Dollar verbuchen, obwohl schon Tausende von Arbeitsplätzen abgebaut worden waren. 2010 übernahm der Marktführer für kommerzielle Datenbanken, *Oracle*, Sun und damit auch *MySQL* und *Java*. Als direkte Auswirkung haben sich zahlreiche Open-Source-Projekte von *Oracle* abgewendet. Das Sun-Betriebssystem fristet als *OpenSolaris* sein Dasein.

Linus und Linux

Wie in Teil I dieser Serie erwähnt, wurde 1983 das *GNU*-Projekt gegründet, das zum Ziel hatte, einen komplett freien Unix-Klon zu entwickeln. Das Projekt brachte Compiler und zahlreiche Tools heraus, die ja auch maßgeblich zum Erfolg von *Cygnus* beitrugen. Nur fehlte immer noch der Betriebssystemkern. 1990 wurde mit der Entwicklung des Kerns auf Basis des *Mach*-Mikrokerns begonnen und *The Hurd* genannt. Die Entwicklung ist bis heute noch nicht abgeschlossen.

1987 begann Andrew S. Tanenbaum an der Freien Universität Amsterdam ein unixoides Betriebssystem namens *Minix* als Lehrsystem zu entwickeln. Der *Minix*-Kern war in C geschrieben und ist Teil von Tanenbaums Lehrbuch „Operating Systems – Design and Implementation“. *Minix* entstand unter anderem, weil der Quellcode von *Unix* damals nicht mehr für Lehrzwecke an Universitäten zur Verfügung stand.

Das erste System lief sogar auf PCs mit 8088-Prozessor und 512 KByte RAM. Es enthielt aber schon alle Systemaufrufe der *Unix*-Version 7. *Minix* hätte damals eine entscheidende Rolle unter den Betriebssystemen spielen können. Die Lizenzpolitik von Tanenbaum, der seinen Quellcode zwar offenlegte, aber nicht zur Weiterverwendung oder Abänderung freigab, war jedoch zu restriktiv.

Minix diente einem Informatik-Studenten in Helsinki als Entwicklungsumgebung für seine Experimente mit der damals neuen 80386-Prozessorlinie. Dieser Student, Linus Torvalds, entwickelte aber dann einen voll funktionsfähigen Kernel mit virtuellem Speicher und Speicherschutzmechanismen. Er nannte ihn *Linux*. Die *Minix*-Kommandos und der C-Compiler wurden nach und nach durch *GNU*-Versionen ersetzt.

Die allerersten Teile des *Linux*-Kerns wurden im September 1991 über das Internet freigegeben. Es war wohl genau der richtige Zeitpunkt für ein neues *Unix*. Innerhalb kürzester Zeit fanden sich weltweit Programmierer, die an der Idee Interesse hatten und Erweiterungen programmierten: etwa ein verbessertes System zur Dateiverwaltung, Treiber für diverse Hardware-Komponenten oder Zusatzprogramme wie den *DOS*-Emulator. All diese Einzelkomponenten wurden ebenfalls kostenlos unter der *GPL* zur Verfügung gestellt. Das Gesamtsystem wuchs mit einer atemberaubenden Geschwindigkeit. Die Entstehung dieses neuen Betriebssystems wäre ohne die weltweite Kommunikation der Programmierer via Internet unmöglich gewesen.

Und schon gab es auch Streit um *Linux*. 1992 griff Andrew S. Tanenbaum *Linux* wegen des aus seiner Sicht veralteten Designs an. Später nahm Tanenbaum *Linux* sogar gewissermaßen in Schutz, als Ken Brown in seinem Buch „Samizdat“ schrieb, *Linux* sei nur eine Kopie von Tanenbaums *Minix*: Tanenbaum erwiderte, *Linux* habe ein zu schlechtes Design, als dass es von *Minix* abgeschrieben sein könne.

Ein wesentlicher Faktor dafür, dass *Linux* frei von den Rechten der Software-Riesen ist und dennoch derart schnell entwickelt werden konnte,

war die zu diesem Zeitpunkt schon frei verfügbare GNU-Software (und all das, was Unix-Entwickler so in der Schublade hatten). Linux ist keineswegs aus dem Nichts aufgetaucht, wie das manchmal fälschlich dargestellt wird, sondern baut auf einer breiten Basis freier Software auf – was die Verdienste von Linus Torvalds um den Kernel aber nicht schmälert. Noch heute, 22 Jahre später, hält er die Entwicklergemeinschaft zusammen (was nicht immer eine leichte Aufgabe ist). Er ist dafür bei der gemeinnützigen *Linux Foundation* angestellt. Doch auch die GNU-Tools sorgten dafür, dass aus einem Kernel recht schnell ein vollständiges System wurde, das dann für eine noch größere Entwicklergemeinschaft zu einer attraktiven Umgebung werden konnte.

Der Name „Linux“ war übrigens nicht die erste Wahl. Dem Administrator des FTP-Servers, auf dem die Linux-Dateien lagen, gefiel keiner der von Linus vorgeschlagenen Namen (etwa „Freax“ oder „Buggix“). Deshalb nannte er das Verzeichnis einfach Linux. Linus gab seinen Widerstand gegen den neuen Namen bald auf und gestand ein, dass Linux einfach der bessere Name war. Bis heute versucht der GNU-Gründer Richard Stallman (Abbildung 3), wegen des hohen Anteil an GNU-Tools den Namen auf „GNU/Linux“ zu erweitern. Das GNU-Projekt und das Debian-Projekt folgten diesem Vorschlag, die meisten anderen Linux-Distributoren lehnen es ab – einerseits aus Bequemlichkeit, andererseits, weil inzwischen eine beachtliche Menge der Linux-Software nicht mehr aus dem GNU-Projekt stammt.



Abbildung 3: GNU-Gründer Richard Stallman 2008 an der Cambridge University. (Bild: Stephen Birch)

Die Kombination aus dem Linux-Kernel, den GNU-Komponenten, der Netzwerk-Software des BSD-Unix, dem ebenfalls frei verfügbaren *X Window System* des *Massachusetts Institute of Technology* (MIT) und dessen Portierung *XFree86* für PCs

mit Intel-Prozessoren sowie aus zahlreichen weiteren Programmen macht eine Linux-Distribution zu einem kompletten Unix-System. An dieser Stelle gehört eine sprachliche Feinheit angebracht. Linux ist streng genommen nur der Kernel des Betriebssystems. Dieser und das ganze Drumherum an Tools, grafischer Oberfläche, Installations-Skripten, Update-System und Service ergibt erst eine Linux-Distribution – und davon gibt es inzwischen für jeden Geschmack das Passende.

Linux-Distributionen

Eine der ersten Distributionen nannte sich *SLS* (Softlanding Linux System) und erschien 1992. Im gleichen Jahr folgte *Yggdrasil Linux* (Yggdrasil ist die Weltesche aus der nordischen Sagenwelt). Die älteste bis heute existierende Distribution, *Slackware* von Patrick Volkerding, folgte 1993 und basierte auf SLS. Die Slackware-Version mit Kernel 0.98 (damals auf 30 Disketten zu 3,5 Zoll erhältlich) war auch das erste Linux, das der Autor dieses Artikels verwendet hat. Seither wuchsen die Distributionen kontinuierlich an Umfang, und die Installation wurde immer einfacher. Inzwischen werden manche Distributionen auch immer Windows-ähnlicher, wobei nicht den Komfort eines Desktop-Environments oder einer fast automatischen Installation gemeint ist, sondern die Entmündigung des Users in punkto Administration. Auf *Ubuntu*-Systemen neueren Datums lassen sich manche Aufgaben nur noch mühsam auf der Shell-Ebene lösen, Klicki-Bunti fast wie bei Windows ist erforderlich. Aber vielleicht wird der Autor dieser Zeilen einfach zu alt.

Zurück zu den Distributionen, deren Zahl immer noch zunimmt. Deshalb nur einige Meilensteine: Heraus ragt *Debian* mit einer großen Software-Auswahl. Sie unterstützt zwölf Prozessor-Architekturen und damit mehr als jede andere binäre Linux-Distribution. Zugleich dient sie als Basis vieler Derivate, darunter das sehr beliebte *Ubuntu*-Linux (das selbst schon wieder Basis für andere Distributionen ist). Unter den Debian-Derivaten ist auch zu nennen der – in den Augen des Autors unübertroffene – King der so genannten Live-CDs: *Knoppix*. Einziger Nachteil von Debian ist, dass die enthaltenen Programmversionen oft sehr konservativ sind und man gegebenenfalls aktuelle Versionen nachinstallieren muss.

Arch Linux ist eher etwas für Profis, denn die Installation erfolgt ohne grafische Oberfläche. Diese unabhängige Linux-Distribution hält sich strikt an

das KISS-Prinzip: „Keep It Simple, Stupid“. Dazu gibt es Rolling Releases, also Releases jeweils nach dem aktuellen Bestand der Paketarchive. *Gentoo* ist eine quellbasierte Linux-Distribution ebenfalls für fortgeschrittene Linux-Benutzer, die ihr System komplett individuell einrichten möchten. Bei der Installation werden Kernel und Programme direkt aus dem Quellcode erzeugt.

Red Hat Linux (RHL) wurde von 1994 bis ins Jahr 2003 von der Firma *Red Hat* entwickelt. Der weit verbreitete Paketmanager *RPM* stammt ursprünglich aus *Red Hat Linux*. Neben etlichen Derivaten spalteten sich zwei Versionen ab: *Fedora* ist der direkte Nachfolger des freien *Red Hat Linux*. Er verwendet ausschließlich Pakete mit vollständig freien Lizenzen. *Red Hat Enterprise Linux* (RHEL) ist die kommerzielle, kostenpflichtige Distribution der Firma *Red Hat*. Sie wird oft auf Servern eingesetzt und in den USA häufig Standard: Es heißt dann, „läuft auf RHEL“, anstatt „läuft auf Linux“.

1992 wurde in Nürnberg die *S.u.S.E. GmbH* (Gesellschaft für Software- und Systementwicklung mbH) gegründet. Sie verkaufte Linux-Disketten im Paket, denn das Herunterladen aus dem Internet war damals nur wenigen möglich. Daneben erschien bereits vorher die *Slackware*-Distribution als deutschsprachige Version. Auch die später erscheinende *SuSE Linux*-Distribution war übersetzt, weshalb sie eine große Anhängerschaft im deutschsprachigen Raum fand. Auch wurde das Installationstool bald durch das hauseigene *YaST* von *Suse* ersetzt. Ab April 1994 wurde die Distribution nicht mehr auf Disketten (über 70 Stück), sondern auf CD-ROM ausgeliefert. Noch später gab es dann ein dickes deutsches Handbuch dazu. 2004 kaufte *Novell* die *Suse GmbH*. Das war aber keineswegs das Ende: Es folgte das *openSUSE*-Projekt. 2006 wurde auch der Name der Distribution auf *Opensuse* geändert, um den Einfluss des Projekts widerzuspiegeln und Verwechslungen mit den kommerziellen Ablegern zu vermeiden.

2011 wurde dann *Novell* wiederum von *Attachmate* gekauft. Lediglich einige Patente von *Novell* gingen an ein von *Microsoft* geführtes Konsortium namens *CPTN Holdings*. *Suse* wird in Zukunft wieder als eigene Sparte geführt. Für das *Opensuse*-Projekt soll dies keine unmittelbaren Veränderungen mit sich bringen, hieß es. Das Wichtigste: Die Rechte an der Marke *UNIX* wurden an die *Open Group* übertragen.

Das Ende der Geschichte – vorläufig

Die Geschichte von *Unix* ist die einer ganzen Familie von Betriebssystemen. Besonders ist, dass in der Vergangenheit die freie Verfügbarkeit eines *Unix*-Systems stets Antrieb für die Fortpflanzung der Familie war. Der Erfolg des Ur-Systems von *Ken Thompson* und *Dennis Ritchie* wäre ohne die Mithilfe seiner Benutzer nicht möglich gewesen. Schließlich führen heute *GNU* und *Linux* und die *BSD*-Nachfolger, unterstützt durch zahlreiche Unternehmen, dieses Entwicklungsmodell sehr erfolgreich fort. Selbst *Microsoft*, das offene Entwicklung früher nur als schädliche Konkurrenz betrachtete, beginnt wirtschaftliches Potenzial in einer Beteiligung an einzelnen Entwicklungen zu sehen.

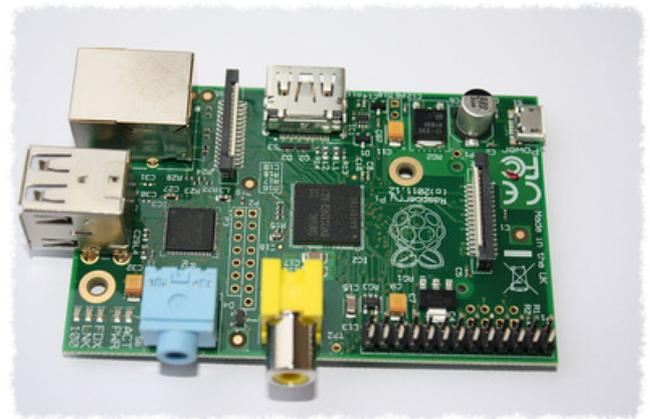


Abbildung 4: Ein Board namens *Raspberry Pi*. (Bild: Philipp Bohk)

Eines der nächsten großen Gebiete für *Unix* sind *Embedded*-Anwendungen (zum Beispiel *Android* oder *DSL-Router*), dann weiterhin *Webserver*, *Datenbankserver*, *Fileserver*, *Clustering* (Hochverfügbarkeit) und in geringerem Maß auch der *Desktop*-Bereich (etwa mit *Mac OS X*). Eine der aktuellen Entwicklungen ist der *Raspberry Pi* (Abbildung 4), ein kreditkartengroßer Einplatinen-Computer, entwickelt von der *Raspberry Pi Foundation*. Die Platine enthält einen 700-MHz-Prozessor mit *ARM*-Architektur und 512 MByte Arbeitsspeicher. Das aktuelle Modell B besitzt eine *Ethernet*-Schnittstelle, zwei *USB*-Anschlüsse und diverse Ein-/Ausgabelösungen für Steuerungsaufgaben. *Linux* für die *ARM*-Architektur kann installiert werden. Statt einer Festplatte werden *Speicherkarten* (*SD* oder *MMC*) als nicht-flüchtiger Speicher verwendet.

Damit steht das Tor zum Internet der Dinge für *Unix* offen.

Chronik

- 1990 SUN veröffentlicht *Solaris 1*
- 1991 *UnixWare*
- 1991 *Linux*
- 1992 *SuSE Linux*
- 1994 *OpenBSD, NetBSD, FreeBSD*
- 1995 *Windows 95*
- 1995 SUN *Microsystems* erfindet *Java*
- 1998 *Cygwin*
- 1999 US-Präsident Bill Clinton ehrt Ken Thompson und Dennis Ritchie für ihre Arbeit
- 2002 *UnixWare 7*
- 2004 *Novell* kauft die *S.u.S.E. GmbH*

- 2010 *Oracle* übernimmt Sun (sowie Java und MySQL)
- 2011 *Attachmate* kauft Novell
- 2012 *Raspberry Pi*

Das Letzte

Etliche Asteroiden bekamen Namen von prominenten Software-Entwicklern oder Open-Source-Projekten:

- (9885) Linux
- (9793) Torvalds
- (9965) GNU, und
- (9882) Stallman.

May the source be with you, but remember the KISS principle!

Literatur

[1] Jürgen Plate: Geschichtsstunde. IT-Frühzeit von 1950 bis 1975. Uptimes 2012-2, S. 22 - 26:

<http://www.guug.de/uptimes/2012-2/index.html>

[2] Jürgen Plate: Geschichtsstunde II. 1975 bis 1990: Unix erobert die Welt. Uptimes 2012-3, S. 30 - 36:

<http://www.guug.de/uptimes/2012-3/index.html>

Die Geschichte von BSD-Unix:

<https://www.bsdwiki.de/Spezial:Verweisliste/Geschichte?title=Geschichte>

SCO vs. Linux – Die unendliche Geschichte:

<http://www.heise.de/ct/artikel/SCO-vs-Linux-Die-unendliche-Geschichte-302076.html>

Das Allerwelts-Unix – 10 Jahre NetBSD:

<http://www.heise.de/newsticker/meldung/Das-Allerwelts-Unix-10-Jahre-NetBSD-76625.html>

Viele Aufgaben und Benutzer – Unix wird 40:

http://www.heise.de/artikel-archiv/ix/2009/08/096_Wortspiel

Cygwin: <http://www.cygwin.com>

Andy Tanenbaum, LINUX is obsolete:

<http://groups.google.de/group/comp.os.minix/msg/f447530d082cd95d>

Linux-Distributionen:

http://de.wikipedia.org/wiki/Liste_von_Linux-Distributionen

Raspberry Pi: <http://www.raspberrypi.org>

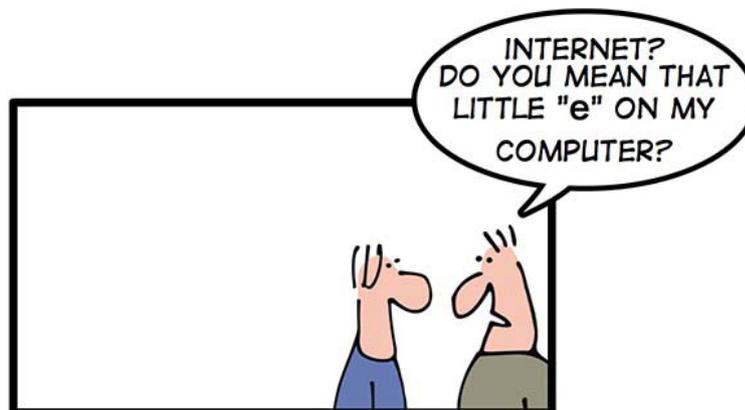
THE classic Unix horror story:

<http://www.lug.wsu.edu/node/414>

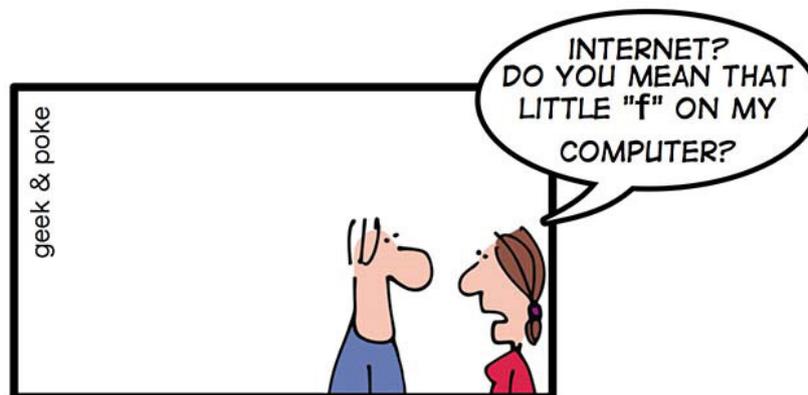
Über Jürgen



Jürgen Plate ist Professor für Elektro- und Informationstechnik an der Hochschule München. Er beschäftigt sich seit 1980 mit Datenfernübertragung und war, bevor der Internetanschluss für Privatpersonen möglich wurde, in der Mailboxszene aktiv. Unter anderem hat er eine der ersten öffentlichen Mailboxen – TEDAS der mc-Redaktion – programmiert und 1984 in Betrieb genommen.



1999: YOUR DAD



2011: YOUR KID

#ffg2013 GUUG-Kongress zu Gast in Frankfurt

Acht Tutorien und 37 Vorträge: Das Frühjahrsfachgespräch 2013 überzeugte durch ein dicht gepacktes Programm auf hohem Niveau. Die rund 170 Teilnehmer freute es.

von Corina Pahrman

Es ist ein fester Termin im Kalender: Ende Februar eines jeden Jahres treffen sich IT-Profis aus den Bereichen Unix, Netze und IT-Sicherheit zum Frühjahrsfachgespräch der GUUG. 2013 gastierten wir an der Fachhochschule Frankfurt/Main.

Wie immer startete das FFG mit zwei vollen Tagen Tutorium, daran angeschlossen folgten zwei Tage Vorträge. Am ersten Vortragstag liefen erstmalig sogar drei Tracks parallel. Das damit erreichte Plus von sieben Vorträgen wurde sehr gut aufgenommen (Abbildung 1). Für genügend Austausch unter den Teilnehmern und Referenten sorgte auch das Social Event – ein geselliger Abend, der in der Friedberger Warte stattfand.



Abbildung 1: Das FFG bietet viel Raum zum Austausch – in den Kaffeepausen wie auch beim Social Event, das immer am Abend des dritten Konferenztags stattfindet. (Bild: Corina Pahrman)

Unter die Motorhaube schauen

Im Mittelpunkt der Konferenz standen die Themen Cloud Computing, Hochverfügbarkeit, Monitoring, Dateisysteme und Netzwerksicherheit. Nach zwei intensiven Tutoriumstagen eröffnete

Aus einer großen Gesellschaft heraus
Ging einst ein stiller Admin nach Haus.
Man fragte: Wie seid Ihr zufrieden?
Wäre's Skripte, sagt er,
ich würd sie ausprobieren.

Frei nach J.W. Goethe, Gesellschaft

der Open-Source-Pionier Kurt Garloff mit einer Keynote den Vortragsteil (Abbildung 2). Garloff – jahrelang bei Suse/Novell als Entwickler, später leitend in verschiedenen Positionen und aktuell als VP Engineering Cloud Services bei der Deutschen Telekom beschäftigt – erklärte dem Auditorium seine persönliche Faszination für Quelloffenheit: „Ich war das Kind, das immer alle Geräte aufschraubte. Und genauso schätze ich es heute, unter die Motorhaube schauen zu können. Und ich mag die Community, die Diskussion, die sich unter den Teilnehmern von Developer Groups entwickelt.“



Abbildung 2: Kurt Garloff erinnerte in seiner Keynote nochmal an die vielen Vorteile von Cloud Computing (Bild: Erwin Hoffmann)

Ein wichtiges Argument für den Einsatz von Open Source sei Garloffs Überzeugung nach jedoch die Unabhängigkeit von Konzernen: „Technologie bestimmt inzwischen unser Leben. Und die grundlegenden Technologien sollten unter der Kontrolle vieler stehen, und nicht unter der eines einzelnen Unternehmens mit rein wirtschaftlichen Interessen.“ – eine Einschätzung, die bei den Zuhörern für große Zustimmung sorgte.

Das FFG als berufliche Weiterbildung

Das Besondere am FFG: absolute Praxisbezogenheit. Sowohl Referenten als auch Teilnehmer tauschten sich aktiv über die Herausforderungen ihres beruflichen Alltags aus: Wie schützt man große Serverlandschaften? Wie spielt man Updates ein, ohne den kompletten Server neu starten zu müssen? Wie überwacht man große Netzwerke, wie wertet man Logfiles aus (Abbildung 3)? Ein Großteil der Vorträge endete nicht mit einer Folie, sondern schloss zum Beispiel eine Demo am offenen Kernel-Fenster an. Eine Sonderrolle nehmen natürlich die Tutorials ein, für die die Referenten eigens virtuelle Landschaften zum Testen angelegt hatten.



Abbildung 3: rsyslog-Autor Rainer Gerhards ging auf die Herausforderungen richtigen Loggings ein. (Bild: Corina Pahrman)

„Das Frühjahrsfachgespräch wird seit Jahren von den Teilnehmern geschätzt, weil es eben mehr als eine IT-Konferenz ist“, so die Einschätzung von

Dirk Wetter, der sich seit vielen Jahren um das Programm des FFG kümmert. „Gerade zu aktuellen Fragen gibt es immer Referenten, die weit vorn sind und diese Perspektive gerne teilen. Daher nimmt man beim FFG in puncto Weiterbildung immer gut was mit.“

Was Ihr tun könnt?

Außerdem ist das FFG auch der Hauskongress der GUUG – eine Gelegenheit, andere Mitglieder zu treffen, an der Mitgliederversammlung teilzunehmen und Vereinsluft zu schnuppern. So ging's 2013 auch verstärkt um die (Neu?-) Ausrichtung der GUUG. In einer Birds-of-Feather-Session, souverän von GUUG-Mitglied Ulf Rudolf moderiert, diskutierten die Teilnehmer verschiedene Wünsche und Visionen. Das war kontrovers. Aber auch fruchtbar. Ideen gibt es unter den GUUGlern sehr viele und sehr gute. Jetzt liegt es an jedem einzelnen, diese auch weiterzubringen. Daher der Aufruf: Die GUUG ist Euer Verein. Bringt Euch ein und füllt ihn mit Leben!

Wenn Euch das FFG gefallen hat, berichtet Euren Kollegen davon. Verdeutlicht den Weiterbildungsaspekt der Konferenz. Seid Ihr Spezialist in Eurem Berufsfeld? Hattet Ihr in den letzten Monaten ein spannendes Projekt, von dessen Erfahrungen auch andere profitieren könnten? Dann überlegt selbst, eine Session oder ein Tutorium zu halten. Der CfP startet in Kürze. Seid Ihr Mitglied in der GUUG? Dann engagiert Euch, besucht Eure regionalen Gruppen, gründet gegebenenfalls selbst Stammtische.

Und – save the date: Das nächste Frühjahrsfachgespräch findet voraussichtlich Ende Februar 2014 in Berlin statt!



Verbatim – gesagt, gehört, geschrieben

„Nur so komische Typen wie ich beschäftigen sich gerne mit dem Loggen.“ (Rainer Gerhards in seinem Vortrag *Strukturiertes Logging mit rsyslog*)

„Diese frontl-Locks sind wie rote Ampeln in Paris. Sie haben reinen Hinweisscharakter.“ (Volker Lendecke in seinem Vortrag *ctdb performance bottlenecks*)

„Patches bedeutet: Man tauscht bekannte Fehler gegen unbekannte ein.“ (Udo Seidel in seinem Vortrag *Reboot reloaded – Linux-Kernel patchen ohne Reboot*)

Protokolliert von Corina Pahrman. Anmerkung der Redaktion: Wer was Schlaues, Dummes, Witziges hört und weitertragen möchte, mailt den Spruch mit „Gesagt, gehört, geschrieben“ im Betreff an die Redaktion (<redaktion@uptimes.de>). : -) Wenn Ihr wisst, wer es gesagt hat, prima – auf jeden Fall aber bitte angeben, wann und wo Ihr es gehört habt.

Links

Rückblicke zu den einzelnen Konferenztage im GUUG-Blog:

<http://blog.guug.de/category/ffg/>

Slides einzelner Vorträge findet Ihr unter Nachlese auf der GUUG-Website:

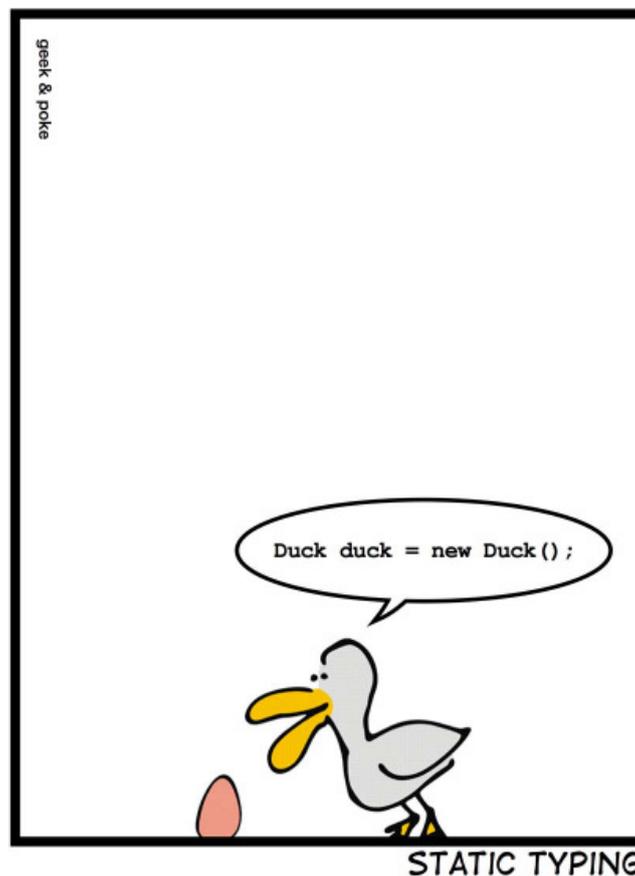
<http://www.guug.de/veranstaltungen/ffg2013/nachlese.html>

Über Corina



Corina Pahrman ist freie Journalistin und Buchautorin. Seit dem FFG 2011 in Weimar unterstützt sie das FFG mit Pressearbeit und füttert das GUUG-Blog mit Berichterstattung direkt vom FFG. Dabei wünscht sie sich für die nächsten Jahre, dass noch mehr Unixer (und andere) die offene, kollegiale Atmosphäre des FFG kennenlernen – und dies immer schön weiterzählen ;-). Wenn Corina nicht gerade Tastatur oder Smartphone malträtiert, verbringt sie Zeit mit ihrer Tochter oder fährt durch die Gegend – gerne zu Konzerten mit gaaanz analoger Gitarrenmusik (leider nicht allzu häufig, siehe ->Arbeit und ->Tochter).

SIMPLY EXPLAINED



Bildernachlese GUUG-Empfang 2013

Der traditionelle GUUG-Empfang im Rahmen des LinuxTag: Wie sah er im Mai 2013 aus? So!

von Anika Kehrer



Abbildung 1: Abendstimmung vor der Alten Pumpe in Berlin-Mitte, südlich vom Tiergarten am Lützowufer des Landwehrkanals. Mit dabei: Michael, Patrick und Axel (rechts oben). Im Gespräch, aufs Essen wartend: Wolfgang, Ingo und Nils (rechts mittig). Bernd heißt vielleicht gar nicht Bernd, und narrt die Welt schon seit Jahren (rechts unten). (Bilder: Anika Kehrer)



Abbildung 2: Auf ins Getümmel! Frank, Mirko und Karsten (links oben) gefällt's. Johannes, Kerstin und Wolfgang (rechts mittig) sehen das bestimmt auch so. Beim Einzel-Shooting bleibt Frank gelassen (links unten). Anders Karsten, der wohl sagen will (rechts unten): „Juhu, hier bin ich, und mit mir ganz viele!“ (Bilder: Anika Kehrer)



Abbildung 3: „Ich soll auch? Na gut...“: Ja, Mirko soll ebenfalls einzeln in der Uptimes (links oben). Johannes und Frank H. (links mittig) hecken bestimmt was aus – gerade wenn Letzterer mit Blicken abzuwiegeln scheint: „Wir? Ach komm“. Martin und Marc hingegen (rechts unten) setzen das Poker-Face auf. (Bilder: Anika Kehrer)



Abbildung 4: Gute Unterhaltung am Tisch von Jana, Moss, Harald und Holger (links oben). Genauso wie bei Werner und Thomas (links oben): So sieht aus, wer zufrieden ist. „Also wirklich, was hast *du* denn gedacht“, scheint Kurt links neben Johannes zu fragen (Mitte). Allein, was Dieter Nils steckte, bevor dieser die Lippen verschloss (links unten), bleibt deren Geheimnis. Nur die Pumpe, die Alte Pumpe weiß es (rechts unten). So wie jeder, der da war und nächstes Mal da sein wird. Auf bald! (Bilder: Anika Kehrer)

Himmel hilf! Ich will doch nur von A nach B!

Oder: Früher waren es Autos, heute sind es Katastrophen.

eine Schmähschrift von **Snoopy**

```
#include disclaimer.h
```

Die GUUG hat nichts mit dem zu tun,
was ich denke oder schreibe.
Außer ich denke über die GUUG nach.
Dies ist eine Glosse.
Dramaturgische Überhöhungen sind kein Zufall.

snoopy@snoopix.de

Es gibt im Leben eines Menschen diverse Ereignisse, die einen Schatten auf seine kümmerliche Existenz werfen, ihm seine Winzigkeit vor Augen führen und ihn seine Demut vor dem Universum wiederentdecken lassen, die normalerweise mit Hilfe der Technik-gestützten Hybris vergessen wird. Wer es gerne eine Nummer kleiner hätte, der lässt mich erzählen vom Kauf einer dieser neu-modischen Automobile und dessen Folgen: uneingeschränkter Zugang zu aufdringlich idiotischen User Interfaces.

In unserem Rudel macht die Autos der Mann im Haus, also Carolin, sie guckt ja auch Fußball, samt Fan-Schminke und Vuvuzela. Sie ist unsere Technikerin, kennt sich hervorragend aus mit den diversen Sonderausstattungen, hat ein geniales Gespür für gute Deals und die (überwiegend männlichen) Autohaus-Brunfthirsch-Vertriebler gut im Griff. So kam es, dass nach zirka einem Jahr Wartezeit ein neuer Passat im Carport stand.

Hier sei nun angemerkt, dass moderne Autos nur entfernt etwas mit den Autos „zu meiner Zeit“ zu tun haben. Heutzutage sind Autos nämlich hochkomplexe kybernetische Systeme, vollgestopft mit Elektronik, Sensoren und diverser Schnickschnack. Die „Bajuwarischen Motorisierten Wich...äh, Wechselbälger“ werben gar damit, dass eines ihrer Erzeugnisse mehr Rechenleistung auf die Straße bringe, als die gesamte NASA-Mondmission der 1960er Jahre. Das klingt wirklich beruhigend, ist aber letztlich vollkommen nutzlos: unsere Straßen sind nämlich wesentlich gefährlicher als Mond oder Mars. Einen Roboter zum Mond oder Mars zu schicken ist ein wenig teuer, aber durchaus erfolgreich. Eine belebte Straße heil zu überqueren, schafft jedoch (noch) keine Maschine.

Dafür schaffen die Maschinen aber mich. Dank

der Zunahme des Silizium-assistierten Fahrens bin ich nun nicht mehr nur bei Code-Review oder Online-Banking den gruseligen Programmen pubertierender Java-Junkies ausgesetzt, nein, noch viel schlimmer: sie verfolgen mich jetzt in der Karre. Ich bin und war nie Autonarr, aber die neuen Software-Releases vergällen mir den letzten Rest an Spaß.

Höhlenmenschen auf Rädern

Dabei beginnt es so harmlos: Ich habe ja nicht nur den sozialen Charme eines Cro-Magnon-Menschen, sondern auch so lange Arme. Diese lasse ich gern beim Autofahren ganz entspannt in den Speichen des Lenkrads baumeln. Das ging im alten Passat erstaunlich lange gut. Im neuen Passat allerdings hat man nicht nur ein Lenkrad mit schnöder Hupe, sondern ein multifunktionales Kontrollelement (unter anderem auch mit schnöder Hupe). Unachtsame und unüberlegte Berührungen lösen sofort und hinterhältig irgendeine Funktion aus – oder schlimmer noch: eine Automatik –, die mir die Kontrolle über Fahrzeug, Schicksal und gute Laune entreißt. Frei nach Bill Bryson in „Notes from a Big Country“:

Ein Leihwagen ist das Fahrzeug, welches Du hinreichend schlecht kennst, sodass ein falscher Knopfdruck den Heckscheibenwischer einschaltet und Du ihn für den Rest der Leihdauer nicht wieder aus bekommst.

Niedlich finde ich auch diese urdeutsch-antliche Abkürzerei. Das zentrale kleine Display, welches mich mit verschiedenen Informationen oder Ansagen versorgt, heißt tatsächlich abgekürzt: MFA. Da brauche ich schon von München

bis Nürnberg, bis ich darauf komme, dass das wohl *Multi Funktions Anzeige* heißen soll, wie dann der Blick ins Web bestätigt. So wie in „Raumpatrouille“ (aka Raumschiff Orion) ORB für *Oberste Raum Behörde* oder TRAV für *Terrestrische Raum Aufklärungs Verbände* steht.

Jedenfalls werde ich nach oben genannter Berührung des Lenkrads ganz unsanft bei 160 Stundenkilometern aus dem Schlummer gerissen, weil aus dem Nichts eine laute weibliche Stimme sagt: „Sie befinden sich im Hauptmenü“. Komisch, ich dachte ich befinde mich in einem Auto, nicht in einem Menü. Toll, was es bei VW alles an Extras gibt. Die Stimme leiert weiter: „Sie können anrufen, Sie können Assistenten aktivieren, blabla“.

Ich drücke ein wenig panisch auf dem Lenkrad herum. Auf einmal höre ich: „Sprachsteuerung aktiviert“. Na, das find' ich ja cool. Ich probiere ein paar einfache Kommandos:

„Rechts“

Keine Reaktion.

„Links“

Nichts.

„Schneller“

Immer noch nichts.

Weder die virtuelle Dame, noch der Wagen, zeigen eine Reaktion. Ich gehe aufs Ganze:

„Computer! Starte Selbstzerstörungs-Sequenz Omega. Identifikation Jean-Luc Picard, Autorisierung Alpha-Alpha 3-0-5.“

Wieder nichts. Na gut, denk ich mir, ganz schön clever, dass die Karre peilt, gar nicht von Jean-Luc Picard bedient zu werden. In Wirklichkeit heiße ich nämlich Locutus von Borg...

Die nächsten 30 Kilometer oder so drücke ich auf dem Lenkrad diverse Knöpfe und versuche mit Hilfe von Buffer Overflows und undefinierten Eingaben, die Turing-Maschine des Passats in manische, von den Entwicklern nicht vorhergesehene Zustände zu versetzen. Irgendwann dämmert mir, dass es eventuell keine gute Idee ist, bei 130 die CPU des Autos zu „bricken“. Wer weiß, welche Pakete dann über den CAN-Bus tänzeln und das Fahrgerät unkontrollierbar machen. Ich breche fürs Erste meine Versuche ab.

Dennoch sind Software und Benutzerschnittstellen dieser Autos einfach gruselig. Ich kann einige Funktionen auslösen, die scheinbar nichts tun. Die Namen sind auch lustig. Hilft mir der „Müdigkeits-Assistent“ dabei, müde zu werden? Auch jedes Bluetooth-fähige Handy versucht sich der Wagen gefügig zu machen. Entsprechend versucht er, den *Missratenen Benutzer* einzubinden. Wir Sind Die Borg und Widerstand Ist Zwecklos,

weil Wir Werden Assimiliert... Es wäre eigentlich nicht so schlimm, aber die dappige Handysuche beschäftigt den Wagen derart, dass andere Funktionen minutenlang nicht verfügbar sind. Superklasse.

Creeping Featurism

Die Benutzerschnittstelle des Autos beschränkt sich eben nicht mehr nur auf einen Satz Pedale, Lenkrad und so weiter, sondern bietet ein ganzes Bouquet von Funktionen, die man mit Hilfe der Elektronik nun auch rein mechanisch vollkommen versaubeuteln kann. Das kommt davon, dass nun Programmierer auch für die Mechanik zuständig sind und nicht mehr die zu Hochdeutsch unfähigen, aber technisch hochbegabten Mittelstands-Meister.

[Passender EDV-Witz des Tages: Die beste Ehefrau von allen ruft ihren Programmierer-Ehemann in der Arbeit an: „Schatz, wenn Du heimkommst, bringe bitte aus dem Supermarkt ein Brot mit, und wenn sie Eier haben, bring sechs Stück.“ Er kommt mit den Einkäufen nach Hause. Sie meint erstaunt: „Warum hast Du denn sechs Brote gebracht?“ Er: „Sie hatten Eier.“]

Früher, mein Sohn, meine Tochter, da gab es noch so richtige Handbremsen und die waren so richtige Hebel. Das war praktisch. In der Fahrschule hieß es: „Anfahren am Berg“, und die dafür nötigen Übungen haben sicher so manche Kupplungsscheibe das Leben gekostet. Heute ist das viel zu simpel: Heutzutage ist die Handbremse nämlich wohlfeil ellegdrohnisch und geht automatisch an, wenn Du stehst. Man kann sie auch händisch per Knopfdruck aktivieren, nur dann nicht so einfach am Berg lösen. Denn unsere findigen Ingenieure haben natürlich eine schönere, weil kompliziertere Lösung parat: die Auto-Hold-Funktion. Die hält das Auto automatisch fest, etwa an der Steigung. (Dann rubbelt halt der Rechner die Kupplung kaputt. Irgendeiner muss es ja tun.)

Das ist Fortschritt: aus einer einfachen mechanischen Lösung macht man zwei elektronische. Ist ja viel einfacher so. Ich nenn's einfach nur Spieltrieb.

Ebenso musste man beim Rückwärtsfahren mal in den Rückspiegel gucken oder den Kopf nach hinten drehen (vgl. „Hello, world!“), damit man weiß, wo man hinfährt. Heute gibt es da natürlich Kameras oder, im Fall unseres Passats, diese Proximity-Piepser. So praktisch das ist, so blöd ist es, wenn ich im Radio Bayern 5 höre und dann rückwärts fahren will. Gemäß Murphy lege ich

immer genau dann den Rückwärtsgang ein, wenn gerade eine sehr interessante Reportage im Radio kommt. Schon wird das Radio ganz abgedreht und ich höre nur noch den Piepser.

Als ob ich nicht in der Lage wäre, den Piepser zu beachten, während das Radio läuft. Ich bin die Frau im Haus und daher auch multitaskingfähig. Diese Funktion beleidigt wirklich meine organische CPU.

Atom-Uhren im Atom-Zeitalter

Irgendwann wird es Winter und ich fahre zum Händler zwecks Reifenwechsel. Die nette Dame an der Service-Wabe guckt in ihren Rechner und meint:

„Oh, der Wagen ist ja noch recht neu, wie sind Sie denn zufrieden?“

Sag ich: „Die Karre ist scheiße.“

Die Dame schaut erstaunt: „Wieso, was fehlt denn?“

Ich: „Die Uhr geht falsch.“

Faszinierend, Captain Kirk. Heutige Autos verfügen über mehrere Quellen für eine korrekte Uhrzeit: RDS, das Navi mit GPS und TMC, einige sogar Fernsehen. Aber niemand schafft es, diese Informationen in die Borduhr einzuspeisen. Natürlich muss man die Uhr nicht nur dauernd nachstellen, nee nee, damit sich keiner auf der Kraftwerk-kompatiblen Autobahn langweilt und immer was zu fummeln hat, darf man auch noch selbst von Sommer- auf Winterzeit umstellen. Es gibt auch Cro-Magnon-Entwickler, nicht nur Fahrer...

Warum bekomme ich nicht ein Menü mit Datum/Uhrzeit und kann dann wählen zwischen: Manuell, Automatisch (RDS), Automatisch (NTP), Automatisch (GPS), oder so? Weil eben keiner mehr dran denkt, dass jemand auf die Uhr sehen möchte. Doch warum bauen sie dann überhaupt eine ein, wenn sie derart nutzlos ist?

Needless to say: Das 21. Jahrhundert ist zu mehr als einem Achtel vorbei. Die Uhren gehen falsch.

Vive la France!

Aber eigentlich sollte ich nicht über den Passat meckern. Peugeot macht die verfassungsgemäße Nutzung eines Autos noch schwerer. Da hat wohl jemand bei Microsoft gelernt, um dann bei PSA die Software zu bauen.

Einmal kurz den Scheibenwischer betätigt und schon mahnt die Anzeige: „Automatische Schei-

benwischer aktiviert“. Aber ich wollte doch nur einmal kurz wischen! Also musst Du den Hebel nach ein mal Wischen zunächst nach oben in die erste Raste hebeln und dann wieder herunter. Beruhigend sagt das Display: „Automatische Scheibenwischer deaktiviert“. Wir halten fest: Die triviale Operation des Einmal-Wischens dauert nun dreimal länger, als – nun ja – die triviale Operation. Super! Offensichtlich fahren die Entwickler nur bei Sonnenschein, alles andere wäre ja auch zu blöd.

Unser Freund Hanjo leiht sich ab und zu den Peugeot aus. Er hat es tatsächlich mit seiner Grobmotorik geschafft, die RDS-Informationen des eingebauten Autoradios in das Display zu bringen. Daran bin ich bis dato immer gescheitert. Sagenhaft. Nur tun sich seltsame Lücken auf: das Ding kann nämlich keine Umlaute oder das scharfe S. So lese ich dann: „Bundesverfassungsgericht ber t ber den L nderfinanzausgleich“. Willkommen im 21. Jahrhundert. Da kann man sich natürlich voll auf den Verkehr konzentrieren, wenn man dann während der Fahrt Buchstaben kaufen muss. Dabei habe ich die Menüs und Spracheinstellungen auf Germanisch eingestellt. Nutzt rein gar nichts.

Übrigens heißt die Anzeige dann als Sender: *rot.txt*. Ich glaube, ich weiß, wie die Cro-Magnons das Teil implementiert haben. Mich schauderts. NT - Neanderthal Technology: light fire, build server, make car.

Ich vermute mal, dass das Modell (Preisträger des Goldenen Lenkrads 2007) hingegen fehlerfrei Französisch darstellen kann, sonst würde das vielbelächelte *Ministerium zur Reinhaltung der Französischen Sprache* dem Konstrukteur wohl das Goldene Lenkrad in eine Körperöffnung einführen, welche normalerweise nicht zum Sprechen verwendet wird. Und ein paar mal lenken, um der amtlichen Meinung entsprechend Nachdruck zu verleihen.

Ach ja: Auch im Peugeot geht die Uhr falsch. Das hat man als zahlender Kunde von so strategischen Partnerschaften zwischen Firmen: Dem freien Cro-Magnon-Entwickler-Pappnasenaustausch sind keine Grenzen gesetzt.

Wartung und Ähnliches

Zum User Interface eines Autos gehört, was Dir der nette Hersteller so erlaubt. Dazu gehört auch, mal 'ne Birne zu tauschen. Merke: Der erste Satz hält fast ewig, doch sobald Du austauschst, musst Du alle drei Monate ran. Die Ersatzbirnen gehen nämlich laufend kaputt.

Beim Passat ist das ganz toll, da darfst Du erstmal den halben Motorraum zerlegen und Luftschläuche abschrauben, bis Du an die Lampen rankommst. Beim Peugeot gibt es ganz lustige Klammern, die die Birnen in den Lampen festhalten. Diese schnippen dann bei der leisesten Berührung zur Unzeit weg und entschwinden in den Tiefen des Motorraums. (Dies hat zur Aufnahme eines Video-Endoskops samt Grapsch-Haken und Magneten von Conrad Electronic in Snoopys EDV-Werkstatt geführt).

Ärgerlich: Die Klammern kannst Du nur mit der ganzen Lampe nachbestellen (150 €), das Endoskop waren nur 70...). In der *Cyber-Bucht* gibt es allerdings Leute, die biegen sowas aus Draht und verkaufen das. Zum Glück. Für die Extrabonuspunkte platziert aber Peugeot dann noch einen scharfkantigen Behälter so nah an der Leuchte, dass Du dir beim Lampenwechsel den Handrücken ruinierst. Da lernt man fluchen, sogar auf Französisch.

Ich glaube, beim nächsten Autokauf werde ich einfach den Vertriebler bitten, mit mir mal vorher die Abblendlicht-Birne zu tauschen. Wenn er dafür länger als eine halbe Stunde braucht oder Verletzungen und miese Laune zeigt, wird die Karre nicht gekauft. Finde ich ein gutes Kaufkriterium. Vielleicht tut sich ja was, wenn Du die am Geldbeutel packst.

Coda

Die Digitalisierung der Welt ist an so Allerweltsgegenständen wie Autos gut sichtbar. Analoge Geräte wie Menschen zählen nix (vgl. *Bombe, Neutronen*-). Das Autoradio zum Beispiel hat ja einen Balance-Regler. Ich habe mal in der Tonstudioteknik gearbeitet und möchte eben das Stereo mittig hören. Aber der Balance-Regler ist digitalisiert und gemäß meinem Gehör liegt die Wahrheit zwischen dem zweiten und dritten Balken rechts. Was soll ich also tun? Entweder links ist zu laut oder rechts ist zu laut. Stimmen tut es nicht. Am liebsten würde ich das Radio samt Entwickler dem Händler in die große Schaufenster-Scheibe pfeffern. Ich empfinde solche Bevormundung als Beleidigung und Körperverletzung.

Erstens – okay, das ist nur der Alterszynismus – sehe ich solche Fehler als Zeichen mangelnder Ausbildung der Ingenieure und Software-Entwickler (und meine Kunden singen mir oftmals Klagelieder davon). Zweitens aber hält die Attitüde *Gut ist gut genug* Einzug. Um Lappalien

wie Uhr, Radio oder RDS kümmert sich eben keiner. Die Qualitätskontrolle hat wohl ganz andere Sorgen. Ich denke, die beschäftigt sich damit, weitere Features einzubauen. Man kennt und liest ja so einiges:

- Die Einparkhilfe bei so Bonzenschleudern wie dem Cayenne, die die Parklücke selbst vermisst und dann einparkt, damit Mylady sich nicht den Hals verrenken muss und der Lack keine Kratzer bekommt. (Das ist kein Sexismus: Die weitaus meisten Cayennes werden von Frauen bedient.)
- Erkennung von Verkehrsschildern, damit der Wagen bremst.
- Erkennung von Soft Targets, wie Fußgänger, um auszuweichen. N.B. erkennbare Rentner sind zum Überfahren freigegeben, um den Staatshaushalt zu entlasten. Gibts als Software-Zusatzoption, bei Bestellung bekommst Du dann Rabatt auf die KFZ-Steuer.

Ich finde solche Dinge unheimlich spannend – da kann ich meinem Hacking-Spieltrieb noch lange frönen, weil die Entwickler immer aktiver ins Fahrgeschehen eingreifen:

Schritt 1 – Nimm ein paar alte Schaufensterpuppen aus der kybernetischen Bucht und schiebe sie mit Schrittgeschwindigkeit auf die Straße. Alternativ rolle ein Transparent mit einem Tempo-30-Schild von einer Autobahnbrücke.

Schritt 2 – Lausche dem süßen Schalmeienklang, wenn die modernen Karren schnell bremsen, während die Oldtimer gar nichts peilen und die Symphonie „Allegro Furioso aus der Zerstörung von Blech und Glas“ aufführen.

Schritt 3 – Das Allerbeste daran ist: Bei diesen Methoden hinterlässt Du am Tatort keine Spuren.

Google Glass auf Rädern

Das ist aber alles nur Vorspiel. Die denken und basteln ja schon viel weiter: zum Beispiel WLAN-Kommunikation, bei der sich die Autos etwa über die Verkehrslage unterhalten, ihr Tempo anpassen, Routen neu berechnen, Zapfsäulen-Preise ausloten und vieles mehr. Oder autonomes Fahren ohne Fahrer. Da gibt es schon Prototypen, das ist in der Industrie weit fortgeschritten. Das ist ja auch viel sicherer, wenn kein Mensch mehr am Steuer sitzt. Eben die üblichen Vorteile, mit denen Überwachungstechnik unters somatisierte Volk gebracht wird.

Wie ich in mehreren Interviews lesen konnte, ist die Autoindustrie gut unterwegs, selbstfahrende Autos zu bauen. Lediglich befürchten sie die mangelnde Akzeptanz und Furcht der Fahrer. Wenn ich mir die Software-Qualität so ansehe, kann ich das durchaus nachvollziehen. Darauf spielen sie natürlich nicht an, sondern stattdessen auf die angebliche Angst der Deutschen vor Bevormundung. Haha, das ist nicht lache: Als ob ein DIN-genormter und ADAC-verfallener teutonischer Autofahrer nicht brav alles tun würde, was ihm sein Fahrzeug sagt. Wenn es ein Volk gibt, welches das Lemming-Gen hat (Lemminge sind biologisch in der Spezies der Duckmäuser), dann sind das wir.

Aber zumindest gibt es dann für Hacker und wahre Bösewichte mannigfaltige Möglichkeiten. Ist doch klasse: Mit ein wenig Netz-Know-how kann man die Autobahn vor sich freiräumen und heizen, was das Gaspedal hergibt. Das Klima fiebert mit. Oder als Bösewicht seine Zielperson punktgenau lokalisieren und an einen gewünschten Ort bringen. Verbrechen im Zeitalter der Couch-Potato: *Victim delivered to your door step*. Ich höre schon die Werbe-Slogans.

Und für die Auto-Industrie ist das vollkommen genial: Sie können erst die Karre verkaufen und dann noch für die Nutzungsrechte abkassieren. Zum Beispiel: Wer schneller fahren möchte, zahlt mehr für ein Tempo 200+. Oder kassieren nach Passagier-Anzahl.

Wenn dann schließlich die autonomen Fahrgastzellen überall verfügbar sind, ist auch die Angriffsfläche enorm viel größer und spaßiger. Das ist doch viel besser als Matchbox-Autos oder Mattels „Hot Wheels“. Die ganze Republik wird zur Carrera-Rennbahn.

Leider auch für den kompetent und weise agierenden Staatsapparat: Der weiß nun ganz genau, wer wann wohin unterwegs ist. Der Verfassungsgrundsatz der Bewegungsfreiheit (was übrigens auch klar bedeutet: frei von staatlicher Beobachtung) ist natürlich irrelevant. Geht doch um den Kampf gegen den Terrorismus. Könnte man ja auch das Maut-System für nutzen, aber das verhindert die Autolobby. Und unsere Übermutter Angela tanzt, wenn deren bayerische Blaskapelle aufspielt. Ganz. Von. Allein.

Schicker Kollateralschaden ist dann auch die Kontrolle dieser lästigen investigativen Journaille: Es darf dann niemand mehr gewisse Sperrgebiete ansteuern. Wenn das jemand will, verriegeln

wir einfach die Türen und fahren die Täter gleich in die Lubjanka, dort wird die Eindringlinge niemand finden. Vorher assimilieren wir noch ihre Handys (vgl. Passat), damit sie keine Hilfe rufen und auch nicht geortet werden können. Aber all das geschieht ja nur zu unserem Schutz.

Ich weiß, ich weiß, ich sollte nicht so misstrauisch sein. Schließlich sind unsere Ermittler und V-Männer so integre Menschen, dass alle Bewegungs- und Telefon-Daten vollkommen sicher sind und niemals missbraucht werden. Sie sind ja auch dermaßen beflissen bei der Ermittlungsarbeit, dass sie eine NSU-fahrende Frau Zschäpe auch im biblischen Sinne erkenntnistlich behandeln und der (G)V-Mann mit ihr eine dreimonatige Liaison pflegte. Und er erkannte Beate...

Aber das macht ja auch nichts. Wenn analoge Männer analogen Frauen in die rechtsradikale Venusfalle tappen, dann schicken wir eben einfach ein digitales Phallus-Symbol („Toys for the Boys“) in Form einer Drohne los. Die regelt das dann auf Obama-Art. Vorher fallen einige Akten für den Untersuchungsausschuss in diese blöden, von Urban Priol postulierten, freilaufenden Schredder in den Gängen des Verfassungsschutzes. Und schon macht man weiter wie bisher.

Aufmerksamen Lesern rate ich zu lernen, wie man normale Autos selbst baut. Das ist sicher bald eine gute Idee. Und mit dem Know-how kann man dann echt Geld machen... Man darf nur bei der Kunden-Akquise nicht zu wählerisch sein. Aber es gibt ja Vorkasse.

Hauptsache *die* stimmt.



Hilfreiches für alle Beteiligten Autorenrichtlinien

Selbst etwas für die Uptimes schreiben? Aber ja! Als Thema ist willkommen, was ein GUUG-Mitglied interessiert und im Themenbereich der GUUG liegt. Was sonst noch zu beachten ist, steht in diesen Autorenrichtlinien.

Der Schriftsteller ragt zu den Sternen empor,
Mit ausgefranstem T-Shirt.
Er raunt seiner Zeit ihre Wonnen ins Ohr,
Mit ausgefranstem T-Shirt.

Frei nach Frank Wedekind, Die Schriftstellerhymne

Wir sind an Beiträgen interessiert. Wir – das ist diejenige Gruppe innerhalb der GUUG, die dafür sorgt, dass die Uptimes entsteht. Dieser Prozess steht jedem GUUG-Mitglied offen. Der Ort dafür ist die Mailingliste <redaktion@uptimes.de>.

Formate

ASCII: Am liebsten verarbeiten wir blanke ASCII-Texte (UTF-8).

L^AT_EX: Es ist nicht erwünscht, eigene Layoutanweisungen einzusenden. Wir setzen zwar die Uptimes mit L^AT_EX: Weil wir aber – wie es sich beim Publizieren gehört – mehrspaltig setzen und ein homogenes Erscheinungsbild anstreben, verwenden wir für die Uptimes bestimmte Formatierungen. Wir behalten uns vor, Texte für die Veröffentlichung in der Uptimes umzuformatieren. Eine Vorlage mit den von uns verwendeten Auszeichnungen für Tabellen, Kästen und Abbildungen gibt es per Anfrage unter <redaktion@uptimes.de>.

Listings: Der mehrspaltige Druck erlaubt maximal ca. 47 Zeichen Breite für Code-Beispiele, inklusive 1 Leerzeichen und einem Zeichen für den Zeilenumbruch innerhalb einer Code-Zeile (Backslash). Breitere Listings formatieren wir um, verkleinern die Schriftgröße oder setzen sie als separate Abbildung.

Bilder: Wir verarbeiten alle gängigen Bildformate, soweit ImageMagick sie verdaut und sie einigermaßen hochauflösend sind. Am besten eignen sich aber PNG- oder PDF-Bilddateien. Bitte plant bei längeren Artikeln mit 1 Abbildung pro 3000 Zeichen. Das müssen nicht Bilder sein, sondern auch Tabellen, Listings oder ein Exkurskasten sind möglich. Bitte verseht Eure Bilder nicht mit eigenen Rahmen oder Verzierungen, weil die Redaktion diese im Uptimes-Stil selbst vornimmt.

Themen und Zielgruppe

Die Uptimes richtet sich als Vereinszeitschrift der GUUG an Leser, die sich meistens beruflich mit Computernetzwerken, IT-Sicherheit, Unix-Systemadministration und artverwandten Themen auseinandersetzen. Technologische Diskussionen, Methodenbeschreibungen und Einführungen in neue Themen sind für dieses Zielpublikum interessant, Basiswissen im Stil von *Einführung in die Bourne Shell* hingegen eher nicht. Wer sich nicht sicher ist, ob sein Thema für die Uptimes von Interesse ist, kann uns gern eine E-Mail an <redaktion@uptimes.de> schicken.

Länge

Ein einseitiger Artikel hat mit zwei Zwischentiteln um die 2.700 Anschläge. Mit etwa 15.000 Anschlägen – inklusive 3 Abbildungen – landet man auf rund vier Seiten. Wir nehmen gern auch zum Beispiel achtseitige Artikel, achten dabei aber darauf, dass der Zusammenhang erhalten bleibt und dass es genug Bilder gibt, damit keine Textwüsten entstehen. Wer Interesse hat, für die Uptimes zu schreiben, macht sich am besten um die Zeichenzahl nicht so viele Gedanken – auch für kurze oder lange Formate finden wir einen Platz. Die Redaktion ist auf Wunsch bei der konkreten Ideenentwicklung behilflich. Für eine Artikelidee an <redaktion@uptimes.de> reicht es, wenn Ihr ein bestimmtes Thema behandeln wollt.

Beitragsarten

Neben Fachbeiträgen sind Leserbriefe, Buchrezensionen, Konferenzberichte und Berichte aus dem Vereinsleben interessant. Wer nicht gleich mehr-

seitige Artikel schreiben möchte, beginnt also mit einem kleineren Beitrag.

Manuskripte

Am einfachsten ist es, wenn wir ein Manuskript per E-Mail an <redaktion@uptimes.de> erhalten. Das ist jederzeit möglich, spätestens jedoch vier Wochen vor dem Erscheinen der nächsten Uptimes. Zum eigentlichen Manuskript ist ein kleiner Infotext zum Autor wichtig, ein Bild ist sehr wünschenswert.

Nützlich ist, wenn der Text vor Einsendung durch eine Rechtschreibkorrektur gelaufen ist. `aspell`, `ispell` oder `flyspell` für Textdateien sowie die von LibreOffice bieten sich an. Wenn Ihr Euren Text an die Redaktion schickt, solltet Ihr also weitestmöglich bereits auf die Rechtschreibung geachtet haben: Nach der Einschickung ist Rechtschreibung und Typo-Korrektur Aufgabe der Redaktion. Die Texte in der Uptimes folgen der neuen deutschen Rechtschreibung.

Redaktion und Satz

Wir behalten uns vor, Texte für die Veröffentlichung in der Uptimes zu kürzen und zu redigieren. Das bedeutet, dass der Artikel nicht ausufert, versehentliche Leeraussagen wegfallen, Syntax und Satzanschlüsse geglättet werden, dass Passiva und Substantivierungen verringert und Unklarheiten beseitigt werden (die zum Beispiel Fragen offen lassen oder aus Passivkonstruktionen resultieren, ohne dass der Schreibende das merkt). Manchmal ist dieser Prozess mit Nachfragen an den Autoren verbunden.

Die endgültige Textversion geht jedem Autoren am Ende zur Kontrolle zu. Dabei geht es um die inhaltliche Kontrolle, ob sich durch den Redaktionsprozess Missverständnisse oder Falschaussagen eingeschlichen haben. Danach setzt die Re-

daktion die Artikel. Wenn der Satz weitgehend gegeben ist – also ein *Release Candidate* als PDF vorliegt – erhalten die Autoren als erste diesen RC. Danach wird die Uptimes dann veröffentlicht.

Rechtliches

Die Inhalte der Uptimes sollen ab Veröffentlichung unter der CC-BY-SA-Lizenz stehen, damit jeder Leser die Artikel und Bilder bei Nennung der Quelle weiterverbreiten und auch weiterverarbeiten darf. Bei allen eingereichten Manuskripten gehen wir davon aus, dass der Autor sie selbst geschrieben hat und der Uptimes ein nicht-exklusives, aber zeitlich und räumlich unbegrenztes Nutzungs- und Bearbeitungsrecht unter der CC-BY-SA einräumt.

Bei Fotos oder Abbildungen, die nicht selbst erstellt wurden, flutscht das gern mal durch: Es ist wichtig, dass der Autor sich bei dem Urheber die Erlaubnis zu dieser Nutzung einholt, und fragt, wie die Quelle genannt zu werden wünscht. Die Frage nach der CC-BY-SA ist hierbei besonders wichtig.

An Exklusivrechten, wie sie bei kommerziellen Fachzeitschriften üblich sind, hat die Uptimes kein Interesse. Es ist den Autoren freigestellt, ihre Artikel noch anderweitig nach Belieben zu veröffentlichen.

Finanzielles

Für Fach- und literarische Beiträge zahlt die GUUG dem Autor nach Rechnungstellung durch den Autor pro Seite 50 € zuzüglich eventuell anfallender USt. Beiträge für die Rubrik „Vereinsleben“, Buchrezensionen und Artikel bezahlter Redakteure sind davon ausgenommen. Gleiches gilt für Paper, wenn die Uptimes die Proceedings der Konferenz beinhaltet.



Nächste Ausgabe (digital): Uptimes 2013-3, Winter-Ausgabe

- Redaktionsschluss: Sonntag, 10. November 2013.
- Erscheinung: 13. Dezember 2013.
- Gesuchte Inhalte: Fachbeiträge über Unix und verwandte Themen, Veranstaltungsberichte, Rezensionen, Beiträge aus dem Vereinsleben.
- Artikelideen und Manuskripte an: <kehrer@guug.de> oder direkt an die Mailingliste <redaktion@uptimes.de>

Über die GUUG

Vereinigung deutscher Unix-Benutzer

German Unix User Group e.V.

Die Vereinigung Deutscher Unix-Benutzer hat gegenwärtig rund 700 Mitglieder, davon etwa 90 Firmen und Institutionen.

Im Mittelpunkt der Aktivitäten der GUUG stehen Konferenzen. Ein großes viertägiges Event der GUUG hat eine besondere Tradition und fachliche Bedeutung: In der ersten Jahreshälfte treffen sich diejenigen, die ihren beruflichen Schwerpunkt im Bereich der IT-Sicherheit, der System- oder Netzwerkadministration haben, beim *GUUG-Frühjahrsfachgespräch* (FFG).

Seit Oktober 2002 erscheint mit der *Uptimes* – die Sie gerade lesen – eine Vereinszeitung. Seit 2012 erscheint die *Uptimes* einerseits zu jedem FFG in Form einer gedruckten Proceedings-Ausgabe (ISBN), und andererseits im Rest des Jahres als digitale Redaktionsausgabe (ISSN). Daneben erhalten GUUG-Mitglieder zur Zeit die Zeitschrift *LANline* aus dem Konradin-Verlag kostenlos im Rahmen ihrer Mitgliedschaft.

Schließlich gibt es noch eine Reihe regionaler Treffen (<http://www.guug.de/lokal>): im Rhein-Ruhr- und im Rhein-Main-Gebiet sowie in Berlin, Hamburg, Karlsruhe und München.

Warum GUUG-Mitglied werden?

Die GUUG setzt sich für eine lebendige und professionelle Weiterentwicklung im Open Source-

Bereich und für alle Belange der System-, Netzwerkadministration und IT-Sicherheit ein. Wir freuen uns besonders über diejenigen, die bereit sind, sich aktiv in der GUUG zu engagieren. Da die Mitgliedschaft mit jährlichen Kosten

Fördermitglied	350 €
persönliches Mitglied	90 €
in der Ausbildung	30 €

verbunden ist, stellt sich die Frage, welche Vorteile damit verbunden sind?

Neben der Unterstützung der erwähnten Ziele der GUUG profitieren Mitglieder auch finanziell davon, insbesondere durch die ermäßigten Gebühren bei den Konferenzen der GUUG und denen anderer europäischer UUGs. Mitglieder bekommen außerdem *c't* und *iX* zum reduzierten Abopreis.

Wie GUUG-Mitglied werden?

Füllen Sie einfach das umseitige Anmeldeformular aus und schicken Sie es per Fax oder Post an die unten auf dem Formular angegebene Adresse. Falls Sie die Seite nicht herausreißen wollen: Sie können den Mitgliedsantrag als PDF herunterladen, siehe URL auf dem Mitgliedsantrag.

Impressum

Uptimes – Mitgliederzeitschrift der
German Unix User Group (GUUG) e.V.

Herausgeber: GUUG e.V.

Bruno-Walter-Ring 30

D-81927 München

Tel.: +49-(89)-380 125 95 0

Fax: +49-(89)-380 125 95 9

E-Mail: <redaktion@uptimes.de>

Internet: <http://www.guug.de/uptimes/>

Autoren dieser Ausgabe: Volker A. Brandt, Snoopy, Jürgen Plate,
Wolfgang Stief, Anika Kehrer, Stefan Schumacher, Corina Pahrman,
Kurt Pfeifle

Vi.S.d.P.: Wolfgang Stief, Vorstandsvorsitzender, Anschrift siehe Herausgeber

Chefredaktion: Anika Kehrer

LaTeX-Layout (PDF): Robin Schröder

XHTML-Layout (ePub): Mathias Weidner

Titelbild und -gestaltung: Hella Breitkopf

Bildnachweis: Comicroreihe *geek & poke* CC-by-sa, mit freundlicher Genehmigung von Oliver Widder. Andere Quellennachweise am jeweiligen Bild.

Verlag: Lehmanns Media GmbH, Hardenbergstraße 5, 10623 Berlin

ISSN: 2195-0016

Wenn Sie Interesse an Anzeigen in der UpTimes haben,
wenden Sie sich bitte an <werbung@guug.de>.

Alle Inhalte der UpTimes stehen, sofern nicht anders angegeben, unter der CC-BY-SA.

Alle Markenrechte werden in vollem Umfang anerkannt.

