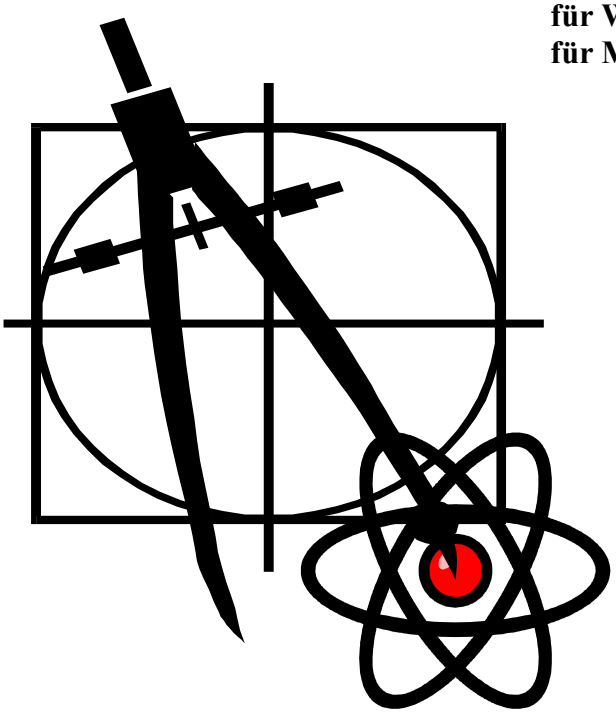
VIERTE DIMENSION

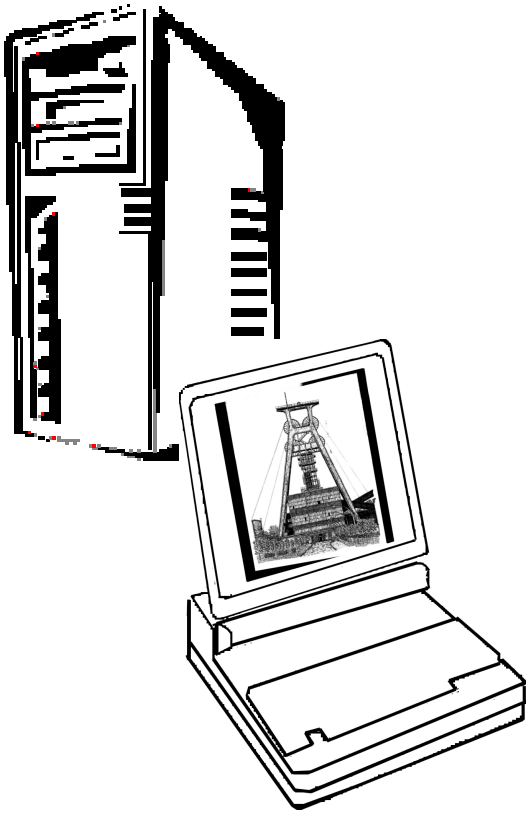
1 / 2005

Das FORTH-Magazin



für Wissenschaft und Technik, für kommerzielle EDV, für MSR-Technik, für den interessierten Hobbyisten.

In dieser Ausgabe:



Dienstleistungen und Produkte fördernder Mitglieder des Vereins

tematik GmbH
Technische Informatik

Feldstrasse 143
D-22880 Wedel
Fon 04103 – 808989 – 0
Fax 04103 – 808989 – 9
mail@tematik.de
www.tematik.de

Gegründet 1985 als Partnerinstitut der FH-Wedel beschäftigten wir uns in den letzten Jahren vorwiegend mit Industrieelektronik und Präzisionsmeßtechnik und bauen z.Z. eine eigene Produktpalette auf.

Know-How Schwerpunkte liegen in den Bereichen Industriewaagen SWA & SWW, Differential-Dosierwaagen, DMS-Messverstärker, 68000 und 68HC11 Prozessoren, Sigma-Delta A/D. Wir programmieren in Pascal, C und Forth auf SwiftX86k und seit kurzem mit Holon11 und MPE IRTC für Amtel AVR.

LEGO RCX-Verleih

Seit unserem Gewinn (VD 1/2001 S.30) verfügt unsere Schule über so ausreichend viele RCX-Komponenten, daß ich meine privat eingebrachten Dinge nun Anderen, vorzugsweise Mitgliedern der Forthgesellschaft e.V., zur Verfügung stellen kann.

Angeboten wird: Ein komplettes LEGO-RCX-Set, so wie es für ca. 230,- € im Handel zu erwerben ist.

Inhalt:

1 RCX, 1 Sendeturm, 2 Motoren, 4 Sensoren und ca. 1.000 LEGO Steine.

Anfragen bitte an

Martin.Bitter@t-online.de

Letztlich enthält das Ganze auch nicht mehr als einen Mikrocontroller der Familie H8/300 von Hitachi, ein paar Treiber und etwas Peripherie. Zudem: dieses Teil ist 'narrensicher'!

Hier könnte Ihre Anzeige stehen!

Wenn Sie ein Förderer der Forthgesellschaft sind oder werden möchten, sprechen Sie mit dem Forth-Büro über die Konditionen einer festen Anzeige.

Secratary@forth-ev.de

Hier könnte Ihre Anzeige stehen!

Wenn Sie ein Förderer der Forthgesellschaft sind oder werden möchten, sprechen Sie mit dem Forth-Büro über die Konditionen einer festen Anzeige.

Secratary@forth-ev.de

KIMA Echtzeitsysteme GmbH

Tel.: 02461/690-380
Fax: 02461/690-387 oder -100
Karl-Heinz-Beckurtz-Str. 13
52428 Jülich

Automatisierungstechnik: Fortgeschrittene Steuerungen für die Verfahrenstechnik, Schaltanlagenbau, Projektierung, Sensorik, Maschinenüberwachungen. Echtzeitrechnersysteme: für Werkzeug- und Sondermaschinen, Fuzzy Logic.

FORTecH Software
Entwicklungsbüro Dr.-Ing. Egmont Woitzel

Budapester Straße 80 a D-18057 Rostock
Tel.: (0381) 46 13 99 10 Fax: (0381) 4 58 34 88

PC-basierte Forth-Entwicklungswerkzeuge, comFORTH für Windows und eingebettete und verteilte Systeme. Softwareentwicklung für Windows und Mikrocontroller mit Forth, C/C++, Delphi und Basic. Entwicklung von Gerätetreibern und Kommunikationssoftware für Windows 3.1, Windows95 und WindowsNT. Beratung zu Software-/Systementwurf. Mehr als 15 Jahre Erfahrung.

Ingenieurbüro
Dipl.-Ing. Wolfgang Allinger

Tel.: (+Fax) 0+212-66811
Brander Weg 6
D-42699 Solingen

Entwicklung von µC, HW+SW, Embedded Controller, Echtzeitsysteme 1-60 Computer, Forth+Assembler PC / 8031 / 80C166 / RTX 2000 / Z80 ... für extreme Einsatzbedingungen in Walzwerken, KKW, Medizin, Verkehr / >20 Jahre Erfahrung.

Ingenieurbüro
Klaus Kohl

Tel.: 08233-30 524 Fax: - 9971
Postfach 1173
D-86404 Mering

FORTH-Software (volksFORTH, KKFORTH und viele PD-Versionen). FORTH-Hardware (z.B. Super8) und Literaturservice. Professionelle Entwicklung für Steuerungs- und Meßtechnik.

Inhalt

Diese Ausgabe der VD wird vermutlich ca. vier bis sechs Wochen nach dem Erscheinen der Druckausgabe im Internet auf der Web-Seite der Forthgesellschaft e.V. als PDF veröffentlicht werden

http://www.forth-ev.de/download.html

Falls Sie die entsprechende Datei dort auch nach diesem Zeitraum vermissen sollten, wenden Sie sich bitte an den Webmaster der Forthgesellschaft.

fep

In der nächsten Ausgabe finden Sie voraussichtlich:

- Interessantes aus der Bastelstube von Rafael Deliano
- Was immer SIE uns schicken
- Berichte von der Tagung der Forthgesellschaft

Editorial

IMPRESSUM

Name der Zeitschrift

Vierte Dimension

Herausgeberin

Forth-Gesellschaft e.V.
Postfach 19 02 25
80602 München
Tel.: (0 89) 1 23 47 84
E-Mail:
SECRETARY@FORTH-EV.DE
DIREKTORIUM@FORTH-EV.DE

Bankverbindung: Postbank Hamburg
BLZ 200 100 20
Kto 563 211 208

Redaktion & Layout

Friederich Prinz
Hombergerstraße 335
47443 Moers
Tel.: (0 28 41) 5 83 98
E-Mail: ***VD@FORTH-EV.DE***

Anzeigenverwaltung

Büro der Herausgeberin

Redaktionsschluß

März, Juni, September, Dezember
jeweils in der dritten Woche

Erscheinungsweise

1 Ausgabe / Quartal

Einzelpreis

4,00 € + Porto u. Verpackung

Liebe Leser,

ein gesundes, friedvolles und glückliches Jahr 2005 wünsche ich Ihnen allen von Herzen. Selbstverständlich wünsche ich Ihnen auch Erfolge bei allem, was Sie tun und lassen. Aber ich bin sicher, daß Sie sich diese Erfolge erarbeiten werden und besondere Segenswünsche an dieser Stelle nicht notwendig sind.
Ein Erfolg, den ich mir nicht erarbeiten kann, ist eine inhaltlich hochwertige und interessante VD für Sie. Dazu bedarf es Ihrer Arbeiten, die leider immer spärlicher hier in der Redaktion eintreffen.
Weil ich mir nicht vorstellen kann, daß Sie „forth-müde“ geworden sind, gehe ich davon aus, daß bei Ihnen noch immer in jedem Jahr Tausende forthiger Zeilen entstehen, die nur ein wenig „in Form gebracht“ werden müssen und dann die VD füllen könnten. Bitte schicken Sie mir diese Zeilen!

Wir werden weniger und älter. Das zeigen die Photos von den Tagungen der Forthgesellschaft ebenso wie die Photos in der Gallerie zum Forth-Day 2004 der FIG im Silicon Valley. Aber es kommen immer wieder junge Leute in die Forthgesellschaft (wie auch in die FIG) und übernehmen Aufgaben, die zuvor von anderen Forthern bewältigt wurden. Der Controller-Verleih ist eine solche Aufgabe. Thomas Prinz hat diesen „Job“ über eine Reihe von Jahren hinweg geleistet. Jetzt hat Carsten Strotmann diese Arbeit übernommen. Beiden sei an dieser Stelle gedankt!

Zum Thema Mitglieder freue ich mich immer wieder, wenn ich an dieser Stelle jemand neues begrüßen darf. ***Rolf Lauer*** aus Schneeberg ist seit dem Januar 2005 Mitglied der Forthgesellschaft. Eine kurze Vorstellung, die er Rolf Schöne bereits geschickt hat, können Sie auf der Seite 29 lesen.

Ich würde mich freuen, auch Rolf Lauer zur Forthtagung in Sachsen persönlich begrüßen zu können. Und ich freue mich schon heute auf die “Forther“ aus allen Teilen Deutschlands, die zur Tagung und zu der Mitgliederversammlung anreisen werden; gerne schon am Donnerstag und zu einem Abstecher nach Meißen oder Dresden.

Ihr

Friederich Prinz

Quelltext-Service

Die Quelltexte in der VD müssen Sie nicht abtippen. Sie können diese Texte auch direkt bei uns anfordern und sich zum Beispiel per E-Mail schicken lassen. Schreiben Sie dazu einfach eine E-Mail an die Redaktionsadresse. *fep*

Die Forthgesellschaft wird durch ihr Direktorium vertreten:

Prof. Dr.	Fred Behringer	Kontakte: Direktorium@Forth-ev.de
Dr.	Ulrich Hoffmann	
Dipl. Inf.	Bernd Paysan	

Betreff: VD
Von: Rafael_Deliano@t-online.de (Rafael Deliano)

...
* Mit Interesse Henry Vinerts Bemerkungen zum antiken Luftabwehrsystem gelesen, da bin ich nämlich selber am Materialsammeln bezüglich Nike Ajax.

Das Leitsystem war vermutlich einer der ersten echzeitfähigen embedded Controller, der in Stückzahlen gefertigt wurde, implementiert als Analogrechner in Röhrentechnik und die CPU ein Container auf der Ladefläche eines LKWs.

Die Ursprünge reichen auf Bells "Director" Flak-Feuerleitsystem des 2. Weltkriegs zurück.

Das automatische Steuerungssystem arbeitet unter erheblichem Rauschpegel. Um zu treffen, benötigt man genau wie beim Tontaubenschiessen einen "Vorhalt", also eine Vorhersage ("Prediction"), wo sich das Ziel zu einem Zeitpunkt in naher Zukunft befinden wird.

An Fragestellungen zu diesen Problemen arbeitete z.B. Norbert Wiener was zum (nicht realisierbaren) Wiener-Filter führte, was wiederum Vorstufe des Kalman-Filters von nach 1960 ist.

Das System hat also nicht nur praktische Erfahrungen geliefert, die später Grundlage ziviler Programme (Apollo) wurden, sondern war auch ein Ausgangspunkt der modernen (State-Space) Regelungstechnik.

Kann ich bei Bedarf nochmal kurz als Leserbrief oder lang 1-2 Seiten mit Fotos als Übersichtsartikel verwursteln. Letzeres würde allerdings bis Dezember dauern.

MfG JRD

Anm. des Redakteurs: Diese Rückschau auf die Wurzeln moderner Echtzeitsysteme wird sicher breites Interesse finden. Auf diesen Beitrag warten wir also gespannt.

Betreff: Forth 200x und CoreForth
Von: Anton Ertl <anton@mips.complang.tuwien.ac.at>

...weitergeleitet von Bernd Paysan an die "Vierte Dimension"

Anton Ertl bittet, die Ankündigungen zu zwei Forth-Projekten auch den Mitgliedern der Forthgesellschaft bekannt zu machen, um diesen zu ermöglichen, sich in die Projekte einzubringen.

fep

...

Es folgen die Ankündigungen:
==========
Das sind zwei Ankündigungen von unterschiedlicher Zielrichtung, eine für die Forth-200x-Aktualisierung des ANS-Forth-Standards, die andere für CoreForth, einen Standard für ganz kleine Forth-Systeme. Wenn Sie an einem der beiden (oder beiden) Projekten interessiert sind, tragen Sie sich bitte in die entsprechende Mailing-Liste ein.

Anton Ertl

---------- Forth 200x ----------

Eine HTML-Version findet man unter

http://www.complang.tuwien.ac.at/forth/ansforth/forth200x.html

Kurze Fassung

Es ist ein neuer Standardisierungsprozess (Forth 200x) zur Aktualisierung des '94-Standards im Gange. Er wird ein formales Standardisierungsdokument zum Ergebnis haben. Vorschläge für Änderungen im '94-Standard sollten das RfD/CfV-Verfahren **<http://www.complang.tuwien.ac.at/forth/ansforth/rfds.html>** durchlaufen, bevor sie auf dem Standardardisierungstreffen diskutiert werden. Es existiert zur Zeit eine Mailing-Liste **<http://groups.yahoo.com/group/forth200x/>** für RfDs/CfVs und für andere mit den Forth-200x-Bemühungen zusammenhängende Dinge. Das nächste Standardisierungstreffen findet am Tag vor der EuroForth 2005 statt, d.h., am 20. Oktober 2005, und zwar in Santander (Spanien). Es ist noch nicht entschieden, ob eine offizielle Standardisierungsinstitution (wie ISO) gebildet wird.

Lange Fassung

Auf der EuroForth 2004 hatten wir einen Forth-2005-Workshop über eine Aktualisierung des Forth-Standards. Ein Bild der Ergebnistafel mit einer Zusammenfassung der wesentlichsten Punkte findet sich unter **<http://www.complang.tuwien.ac.at/anton/euroforth2004/photos/img_1824.jpg>.** Die Teilnehmer entschieden sich für einige Abstimmungen, deren Ergebnisse hier zu sehen sind. Es folgt eine aufbereitete Version:

* Sollen solche Bemühungen überhaupt unternommen werden? Den meisten Teilnehmern schien die Idee zu gefallen.

* Sollen sich die neuen Bemühungen nur mit bereits existierenden Praktiken beschäftigen oder auch mit neuen Ideen?

* Sollen wir das RfD/CfV-Verfahren **<http://www.complang.tuwien.ac.at/forth/ansforth/rfds.html>** zur Erzeugung halbformaler Empfehlungen zur Änderung des Standards verwenden, bevor wir über den neuen Standard entscheiden (Abstimmung: 13 ja - 0 nein - 2 Enthaltungen)?

* Es stellte sich heraus, dass eine Anzahl von Teilnehmern nicht im Usenet lesen. Daher wurde der Vorschlag einer moderierten Mailing-Liste (mit einem öffentlichen Archiv) gemacht und Peter Knaggs bot sich als Moderator an (14 Ja - 0 Nein - 1 Enthaltung).
Die Mailing-Liste wurde sofort eingerichtet: Forth200x - und die Moderation besteht zur Zeit aus einer Anerkennung als Mitglied auf der Mailing-Liste (nur Mitglieder haben das Recht, Beiträge zu veröffentlichen). Mitglied kann man

schnurstracks dadurch werden, dass man einen Antrag an forth200x-**subscribe@yahoogroups.com** stellt, oder über die Mailing-Liste-Homepage **<http://groups.yahoo.com/group/forth200x/>.** Ich weiß noch nicht genau, ob man und wie man einen RfD in comp.lang.forth und parallel dazu gleichzeitig in der Mailing-Liste veranstalten kann, aber ich werde es zumindest für die von mir stammenden RfDs versuchen.

* Sollen wir das Standardisierungsproblem von einer Institution wie ANSI, ISO, IEEE etc. behandeln lassen? Wenn ja, von welcher? Die Meinungen darüber gingen auseinander, aber die meisten schienen der Auffassung zu sein, man solle einfach weitermachen und zunächst einmal zusehen, dass ein neues Standardisierungsdokument entsteht, und die Frage einer Standardisierungsinstitution (wenn sie sich überhaupt stellt) auf später verschieben. Es wurde eine Verschiebung um ein Jahr vorgeschlagen (12 Ja - 0 Nein - 3 Enthaltungen). Eines der Argumente gegen die Einbeziehung von Standardisierungsinstitutionen besteht darin, dass eine solche Institution das exklusive Copyright auf das Dokument für sich beansprucht, so dass selbst die Entwickler des Standards das Kopierrecht und das Recht auf Weiterentwicklung verlieren.

* Formatfragen und Editor des Dokuments. Einige Leute waren dafür, mit einer HTML-Fassung zu beginnen und bei diesem Format zu bleiben, andere waren für MS-Word (was viele heftig ablehnten). Einige schlugen LaTeX vor. Unter den Argumenten für Word ist eines das, dass Word "change bars" unterstützt, was in anderen Dokument-Formaten sehr wahrscheinlich nicht der Fall ist. Schließlich wies jemand darauf hin, dass es der Editor des Standardisierungsdokumentes ist, der mit dem Format des Dokuments zurechtkommen sollte. Anton Ertl bot sich als Editor an (15 Ja - 0 Nein - 0 Enthaltungen). Ein gewisses Problem dabei besteht darin, dass ich auch zum Vorsitzenden des ganzen Unternehmens gewählt wurde. Aber ich denke, dieses Problem lässt sich in Wohlgefallen auflösen, bevor die Rolle des Editors wirksam wird.

* Soll ein Standardisierungstreffen einberufen werden? Wo und wann? Wir entschieden uns für einen Tag vor der nächsten EuroForth (9 Ja - 0 Nein - 7 Enthaltungen), d.h., am 20. Oktober 2005, und zwar in Santander (Spanien). Das Standardisierungstreffen sollte nur Vorschläge behandeln, die das RfD/CfV-Verfahren durchlaufen haben.

* Anton Ertl wurde zum Vorsitzenden ernannt und erhielt den Auftrag, die Forth-Gemeinde (in Gestalt der verschiedenen offiziellen und nicht-offiziellen Gruppierungen, die man kennt) von diesem Unternehmen in Kenntnis zu setzen.

---------- CoreForth ----------

Auf der EuroForth 2004 wurde ein weiterer Workshop abgehalten und ich wurde gebeten, die breitere Forth-Gemeinde zu informieren und sie zur Teilnahme an den diesbezüglichen Bemühungen aufzurufen. Das soll also hiermit geschehen (einiges davon war in comp.lang.forth bereits zu lesen):

Soweit ich die Idee verstanden habe, geht es um die Einrichtung eines Standards für ganz kleine Forth-Systeme, die auf Hardware-Kernen wie dem uCore von Klaus Schleisiek oder dem b16 von Bernd Paysan laufen, so dass sie Code zur Implementierung beispielsweise von TCP/IP übernehmen und mitverwenden können. Ein solcher Standard, so ist es beabsichtigt, sollte kleiner als der Core-Word-Set von ANS-Forth sein. Zum Diskutieren dieses Standards wurde unter **<http://groups.yahoo.com/group/coreForth/>** eine Mailing-Liste eingerichtet.

Man beachte, dass diese Bemühung und die Bemühung um Forth 200x unabhängig voneinander laufen (sie können sich gegenseitig beeinflussen, brauchen das aber nicht zu tun).

Ein Bild von der Ergebnistafel dieses Workshops findet sich unter: **http://www.complang.tuwien.ac.at/anton/euroforth2004/photos/img_1826.jpg**.

Peter Knaggs hat einen Bericht über diesen Workshop angefertigt, **<http://dec.bournemouth.ac.uk/forth/euro/ef04/workshops.html#CoreForth>**, jedoch unterscheidet sich einiges von dem, was er schreibt, beträchtlich von dem, an was ich mich selbst erinnere.

Anton Ertl

Lebenszeichen

Lieber Friederich,

ich habe Dich schon zu lange auf einen Report aus dem Silicon Valley warten lassen und, unglücklicherweise, habe ich immer noch keine Zeit gefunden mich hinzusetzen und einen zu schreiben.
...
Ich ging zum SVFIG-Treffen am 25. September, mit all den Swap-Drachen-Ansteckern, die Du mir gesandt hast und kam mit der Hälfte davon wieder zurück. Diese werden wir auf unserem Forth-Tag Mitte November verteilen. Obwohl nur ungefähr ein halbes Dutzend Forther vorbeikamen, war eine Schachtel Anstecker zum Ende des Tages leer, und Dr. Ting hat ein paar weitere Hände voll dankend entgegen genommen, um sie mit zu seinen taiwanesischen Forth-Freunden zu nehmen. Zufälligerweise haben wir uns beide gefragt, ob die Anstecker in China hergestellt wurden und nun die Runde um die Welt komplett machen.

Anmerk.: Nein, die Anstecker wurden in Deutschland hergestellt. Genauere Angaben kann Martin Bitter hierzu machen.

Natürlich füllte Ting den Vormittag mit einigen weiteren seiner Erfolgsgeschichten der Portierung von eForth auf Analog De-

vices' ADuC7024 MicroConverter. ARM7 eForth ist inzwischen auch komplett.

Ich habe allen von Klaus Schleisieks Einladung zur EuroForth 2004 erzählt. Inzwischen wurde dies auch über die SVFIG e-mail-Liste verbreitet.

Nach dem Mittag erleuchtete LaFarr Stuart uns mit Informationen über die Zahlentheorie, die Karl Friedrich Gauss in 1801 begonnen hat. Ich muß sagen, dies war eine Lektion der ich problemlos folgen konnte, ohne den Faden zu verlieren, wie es mir sonst geht, wenn Computer-Wissenschaftler wie Ting loslegen. LaFarr führte uns über Primzahlen zu den arithmetischen Prüfmethoden des "Rauswerfens von Neunen und Elfen" ("casting out nines or elevens"), an deren Nutzung in der Schule, damals in Deutschland, ich mich nur vage erinnern kann. Auf jeden Fall hat er eine ganze Webseite über dieses Thema und damit sei dazu genug gesagt.

Ich muß gestehen, daß ich zwischen den Vorträgen, denen ich nicht folgen konnte, versuchte mit dem Lesen der VD weiterzukommen. Manchmal erscheint mir das Lesen ohne die Hilfe eines Deutsch-Englischen Wörterbuches sehr analog zu meinen Versuchen, die Vorträge über Computerwissenschaften in Englisch zu folgen, wobei Englisch nicht meine stärkste Sprache ist.

Eine Bemerkung noch, bevor ich für heute aufhören muß: Ich habe Ewald Riegers Lösung des "Viererproblems" in VD1/2004 sorgfältig bewundert und genossen und ich überlege, wie schwer es für ihn sein würde, eine Version zu erstellen, bei der die Quadratfunktion mit einer für den Dezimalpunkt (dem Komma in Deutschland, aber nicht in England) ersetzt wird.

Ich müßte überprüfen, welche Bedingungen für dieses Problem standen, als ich es ursprünglich an Dr. Behringer geschickt habe. Ich bin aber sicher, daß der Dezimalpunkt für mich wesentlich war, um die Lösungen bis zur 50 per Hand auszuarbeiten. Die Quadratwurzel habe ich benutzt, aber ich denke nicht, daß ich eine Quadratfunktion eingeführt habe, indem ich die Quadratwurzel aus 4 als Exponent benutzt habe. Vielleicht kannst Du für uns Yankees und unsere britischen Freunde, einen Durchlauf durch solch eine andere Version machen.

Und wenn es in ZForth gemacht war, lerne ich vielleicht noch ein paar Dinge, die ich schon seit vielen Jahren lernen wollte.

Ich sage "Bye, Bye" fürs erste. Da ich aber so viel auf einen Ritt geschrieben habe, kann es sein, daß mein regulärer Bericht über "Lebendiges Forth im Silicon Valley" ein oder zwei Monate warten muß.

Glueckauf!, *Henry*

Hallo, Friederich!

Ich war letztens in solcher Eile, daß ich vergaß Dir zu sagen, daß Dein Thema "Was ist Intelligenz?" nochmals auf mich zukam, nachdem wir ein paar Bemerkungen darüber ausgetauscht hatten. (Ich glaube, Arthur Koestler nennt solche Übereinstimmungen "Synchronität", aber das französische "deja vu" ist populärer). Wie dem auch sei, meine Frau und ich waren zu einer Buchvorstellung von Jeff Hawkins (Erfinder des Palm Pilot), der die Reihen der Autoren betreten hat und sein neues Buch "Über Intelligenz" vorstellte. Es scheint, daß auch er nicht froh über die Idee ist, daß wir versuchen, eine 'künstliche' Intelligenz zu schaffen, ohne das wir wissen, was eigentlich 'natürliche' Intelligenz ist.

Demzufolge verbrachte er 25 Jahre damit, das Gehirn, speziell den Neokortex, zu studieren. Seine Erkenntnisse und Gedanken dazu hat er in diesem Buch zusammengefaßt, welches hauptsächlich für die allgemeine Öffentlichkeit geschrieben wurde.

Ich habe gerade begonnen, das Buch zu lesen und möchte Dir seine Definition von Intelligenz mitteilen: Intelligenz ist die Fähigkeit des Gehirns, die Zukunft auf Basis von Analogien zur Vergangenheit vorauszusagen." Diese Aussage erscheint auf dem Schutzumschlag, welcher von Reviewer des Herausgebers (www.henryholt.com) geschrieben wurde. Hawkins' eigene Worte zu diesem Effekt sind: "Es ist die Fähigkeit, Vorhersagen über die Zukunft zu machen, die die Schwierigkeit der Intelligenz ausmacht." Ich vermute, daß es da wohl noch mehr geben wird, was ich dann später lesen werde.

Gerade jetzt kam mir ein anderer Gedanke: Erinnerst Du Dich an HolonForth? Als ich Dich fragte, was Holon bedeutet, hast Du mich mit Wolf Wejgaard bekannt gemacht und er machte mich mit Koestler bekannt. Vor ein paar Jahren brach aus verschiedenen Gründen die Korrespondenz zwischen Wolf und mir ab. Weist Du, wie es Wolf heute geht?

Zurück zu meinem Bericht... Deine letzten e-mails sagten mir, daß ich mir keine Sorgen machen soll, wenn ich erst Anfang Dezember einen neuen Bericht schreibe. Aber was ist mit Graeme, oder wer auch immer Forthwrite heutzutage zusammenstellt. Falls Du willst, sende Ihnen die Teile meiner e-mail an Dich, damit sie diese in Ihren Magazinen nutzen können. Weiterhin habe ich keine Kopie an Fred gesandt. Du kannst alles, was Du willst, an ihn weiterleiten. Es besteht die Möglichkeit, daß ich keine formelle Besprechung über das SVFIG-Septembertreffen machen werde. Hier sind zu viele andere Dinge zu tun.

Zufälligerweise habe ich schließlich einige Punkte im Schachwettkampf letztes Wochenende erlangt. Ich hatte Glück und punktete 2 1/2 von 4 Spielen. Sie gaben mir sogar 68 $ Preisgeld (Ich mußte den Preis für den zweiten Platz mit zwei anderen teilen).

Ich denke, ich habe Ting genug Swap-Drachen-Anstecker gegeben um ganz Taiwan damit zu versorgen. Falls er eine Reise vor dem nächsten SVFIG-Treffen macht, werde ich ihn fragen, wie ihnen die Anstecker gefielen. Ich habe immer noch genug davon, um jeden zu dekorieren, der zum Forth-Tag im November kommt.

Denkst Du, daß jemand aus der UK-Forth-Gruppe in der Lage ist, die Variation von Ewald Riegers Problem der vier Vieren in einem einfachen Forth in englisch zu programmieren - mit der Einführung des Dezimalpunktes und dem Streichen der Quadratfunktion? Es wäre etwas, was sich sicher für ihr Magazin lohnt. Vielleicht hat Fred eine Idee dazu?

Lebenszeichen

Ich freue mich, daß Du mit Deinem neuen Auto zufrieden bist. Es klingt so, als ob es Dir mehr weh getan hätte, wenn Dein Motorrad kaputt gegangen wäre. Zumindest wurde niemand verletzt, Friederich. Falls es Dir Zeit spart, antworte ruhig in Deutsch. Ich brauche die Übung auch.

Bis nächstens, alles Gute!, *Henry*

Anmerk..: Henry bezieht sich am Ende seines Briefes auf die Information, daß meine Frau und ich im Urlaub 2004 einen Autounfall in Südfrankreich erlitten haben, der bis auf einen wirtschaftlichen Totalschaden und dem nachlaufenden Ärger mit den üblichen Formalitäten glimpflich abgelaufen ist. Seine Vermutung bezüglich meines Motorrads ist absolut richtig.

fep

Anmerk.: Henrys Wunsch nach einer Überarbeitung von Ewald Riegers Arbeit zum Viererproblem würde ich gerne selbst erfüllen, kann dies aber aufgrund extremer Zeitnot nicht leisten. Kann einer von Ihnen, liebe Leser, diese Zeit aufbringen?

fep

Gentlemen,

laßt uns weiter zum Thema des Austausches von Grüßen zwischen Europa und dem Silicon Valley am 20. November diskutieren. Mein Internet-Anschluß hat gestreikt, und obwohl wir einen beträchtlichen e-Mail-Austausch zwischen unseren Forthern hatten, haben wir keinen einfachen Weg gefunden, um eine Webkonferenz oder ein IRC während der parallelen Aktivitäten unserer entsprechenden Konferenzen zu organisieren. Das Beste, was wir zu dieser Zeit vorschlagen können, ist ein einfacher e-Mail-Austausch mit anhängenden Photos unserer Gruppen.

Dr. Ting wird unser Treffen mit seinem berühmten Grill zum Mittag des Samstag, 20.11. bewirten. Dies sollte etwa 19:00 Uhr in Dagstuhl entsprechen. Charles Moore hat versprochen, bei uns zu sein und einen Vortrag um 15 Uhr zu halten. Kevin Apert sagte mir, daß er seine Digital-Kamera mitbringen wird. Ich denke, damit sollten wir in der Lage sein, einige Bilder zu machen und diese Euch gegen 13 Uhr zu senden, kurz bevor wir die Nachmittagssitzung beginnen.

Gebt uns bitte die Adresse, zu der wir sie schicken sollen. Auf unserer Seite ist die beste Adresse, zu der ihr senden könnt, die von Kevin, also: `forther_at_comcast_dot_net`

Falls Ihr irgendeine Idee habt, was wir sonst arrangieren können, um Grüße auszutauschen, schreibt bitte an Kevin (mit einer Kopie für mich). Inzwischen wünsche ich Euch alles Gute in Dagstuhl. Ich bleibe weiterhin "euer Mann über dem grossen Teich."

Henry

Anmerk.: Ich werde von Brief zu Brief neugieriger auf Dr. Tings Grillkünste. Eines vielleicht nicht mehr all zu fernen Tages, so hoffe ich von Herzen, werde ich eine größere Motorradtour über die Reste der Route 66 machen – und dabei Henry, Dr. Ting und vielleicht noch viele der anderen Forther im Silicon Valley treffen. Bis dahin muß ich von Dr.Tings Barbecue träumen und hoffen, daß wir in diesem Jahr in Sachsen grillen können... Immerhin können wir uns über das Grillen (und andere Freuden) unterhalten und uns in die nächste IRC Sitzung unserer englischen Kollegen „einhängen“.

fep

Greetings, Gentlemen!

Vor zwei Tagen, am 20. November 2004, durch ca. 10.000 km Luftlinie getrennt, trafen sich zwei verschiedene Gruppen von Enthusiasten der Programmiersprache Forth auf ihren entsprechenden jährlichen Tagungen. Dies geschieht nun schon ungefähr ein Viertel Jahrhundert. Die EuroForth-Konferenz im Schloß Dagstuhl, Deutschland, und der Forth-Tag am Cogswell College in Sunnyvale, Kalifonien, marschierten gemeinsam für fast einen ganzen Tag vorwärts, obwohl sie durch eine große Entfernung voneinander getrennt waren. Dieses Ereignis erinnerte mich an einen Ausschnitt, den ich aus der Ausgabe des San Francisco Examiner am Sonntag, den 11. April 1993, aufgehoben hatte. Es geht um die "Frag Dvorak"-Frage- und Antwort-Kolummne im "Computer & Technologie" Abschnitt der Zeitung.

Zu dieser Zeit war Dvorak der Editor des PC Magazine; später einmal hörte ich seine Eröffnungsrede auf einer der Embedded-System-Konferenzen. Wie dem auch sei, in dem zuvor erwähnten Artikel antwortete Dvorak auf die Frage eines Lesers über Programmiersprachen. Er bezeichnete Forth als "eine eigenartige Entwicklungssprache, die einen Kultstatus hat". Damals, in meinem dritten Jahr in der SVFIG, habe ich diesen Kommentar nicht begrüßt. Aber heute, nachdem ich die Definition von "eigenartig" und "Kult" nochmals überprüft habe und nachdem ich die langfristige Hingebung der Forther bezeugen kann, kann ich nicht wirklich Dvoraks Semantik widersprechen. Was ist Eure Meinung?

Um meine "Viertel Jahrhundert"-Behauptung zu unterstützen, laßt mich von Seite 79 der Forth Dimension I/6 (März/April 1980) zitieren: "Das erste Treffen des FORTH MODIFICATION LABARATORY (FORML) wurde am Imperial College, London, am 8.-10. Januar 1980 abgehalten. Vertreter sowohl der Europäischen FORTH Nutzergruppe und der FIG nahmen teil." Aus einer späteren Quelle entnahm ich, daß die zweite FORML-Konferenz im Novermber 1980 in Asilomar in Kalifornien abgehalten wurde. Dieser Ort sah seine letzte FORML 1999, dem Jahr in dem der Hauptorganisator, Rob Reiling, starb.

Um meine Behauptung der "langfristigen Hingabe" zu unterstüzten, will ich die alten Jahrgänge der Forth Dimensions durchschauen und die Liste der Autoren der Artikel der FORML, EuroFORML und EuroForth-Tagungen von 1980 bis 1993 aufstellen. Die im folgenden genannten, die am SVFIG Forth-Tag letzten Samstag anwesend waren, sind entweder noch die ursprünglichen Forther oder aber Langzeit-Anhänger: Chuck Moore, Glen Haydon, Bill Ragsdale, John Cassady, Andy Korsak, Robert Patten, John Rible, Robert Smith, C.H. Ting, John Carpenter, Jeff Fox, John Hall, Dave Jaffe, George Perry, Jay McKnight, Kevin Appert. Da waren natürlich noch andere auf dem Treffen, die ich nicht erwähnt habe, aber nur, weil ich

in Eile bin, einige prominente Namen von der anderen Seite "des Großen Teiches" zu erwähnen, die ebenfalls in der obigen Liste auftauchen: Klaus Schleisiek, Anton Ertl, Ulrich Hoffmann, Peter Knaggs, Wolf Wejgaard.

Da sind sicher noch viele weitere, von denen ich weiß, daß sie nicht zu unserem Forth-Tag kommen konnten und sicher einige weitere, die ich allerdings nicht kenne, die, ich bin sicher, auf der EuroForth Konferenz waren.

Entscheidet selbst, meine Freunde. Ist es Hingabe, der selben Flagge so viele Jahre zu folgen!

Mit ungefähr 35 Teilnehmern zeigte unser Forth-Tag dieses Mal keine Reduzierung der Anzahl "Gläubiger". Im Laufe des Tages kamen neun Personen auf das Podium und alle sprachen darüber, was Sie mit Forth getan haben. Dave Jaffe hat ihre Namen und die Themen schon unter **http://www.forth.org/svfig/** aufgelistet, daher möchte ich dies nicht im Detail wiederholen. Persöhnlich habe ich Jeff Fox' Lektion am meisten genossen, wie man sein 'gesamtes' Gehirn zum Programmieren in Forth benutzen kann, anstatt auf nur eine Seite davon begrenzt zu sein. Indem er Marshall McLuhan zitierte, erklärte Jeff warum Forther, die geneigt sind, rechtsseitig zu denken, von der Mehrheit der Programmierer nicht verstanden werden, die dazu gebracht wurden, in einer auf die linke Hirnhälfte orientierten Gesellschaft zu arbeiten.

Chuck liebt es, in Sierra City zu leben, über der Schneegrenze in den kalifornischen Vorbergen. Dort kann man noch nachts die Sterne sehen und zu Fuß auf dem State Highway in die Stadt (mit 500 Einwohnern) ins Restaurant gehen ohne Furcht, überfahren zu werden. Eine neue Version von ColorForth wird wahrscheinlich in ungefähr sechs Monaten erscheinen und OKAD 2 ist voll beschäftigt, neue Forth-Chips zu entwerfen.

Ich habe schon kurz an Friederich über die vorhergehenden SVFIG-Treffen geschrieben, hatte allerdings nicht viel zu berichten. Im Oktober beschrieb Dr. Ting sein Digital-Oszilloskop-Projekt und Dave Jaffe erzählte über Erweiterungen für einen Auto-Fahrsimulator. Dieser soll zum Test älterer Personen oder solcher mit Hirn-Verletzungen eingesetzt werden, bevor deren Fahrerlaubnis verlängert wird. Im September waren auch ungefähr ein Dutzend Personen da, um Tings vorherigen Erfahrungen im oben erwähnten Projekt und LaFarr Stewart's Präsentation einiger weniger bekannten oder vergessenen Aspekte der Zahlentheorie, wie z.B. der modularen Arithmetik, zuzuhören.

Ich vergaß einige wichtige Dinge vom Forth-Tag zu erwähnen:

1. Dr. Ting speiste unfehlbar alle Truppen mit seinem traditionellen und köstlichen Mittagessen vom Grill.
2. Alle Swap-Drachen-Anstecker vom 20jährigen Jubiläum der Deutschen Forth-Gesellschaft wurden an die glücklichen Empfänger überreicht.
3. Eigenartigerweise war keine Frau zum Treffen anwesend (wie das Gruppenfoto, welches Kevin Apert nach dem Mittagessen schoß, zeigen wird, vorausgesetzt es findet den Weg über den Ozean.)
4. Wir erhielten über e-mail Bilder und Video-Clips aus Dagstuhl ungefähr um die gleiche Zeit. Bitte sagt Klaus Schleisiek, daß hier immer noch ein paar alte Forther sind, die ihn erkannt haben.

Oh, noch eine Bemerkungen aus dem Silicon Valley: Am 20. Oktober gingen drei von uns zu Niklaus Wirths Rede über seine Karriere (oder vielleicht sollte ich sagen: er las seine Autobiografie?) im Museum für Computergeschichte. Es war interessant, aber nicht sehr unterhaltsam. Allerdings bezweifele ich, daß Donald Knuth (der im Auditorium war) das gleiche in Deutsch an der ETH besser gemacht hätte.

Das Jahr mag zu Ende gehen, ehe ich Euch wieder schreiben kann, daher laßt mich jetzt Euch allen frohe und friedvolle Weihnachten und ein Frohes Neues Jahr wünschen.

Henry

Hello, again!

Nachdem ich meine e-mail gestern geschickt hatte, sind inzwischen die Gruppenbilder und Thumbnails der Teilnehmer unseres Forth-Tages unter **http://www.forth.org/svfig/fd2004/photos.html** erschienen.

Ich hoffe, daß die Gruppenbilder ihren Weg zu Klaus' spezieller e-mail-Adresse in Dagstuhl gefunden haben. Wir sind noch dabei, den 31 Personen auf dem Bild ihre korrekten Namen zuzuordnen. Während wir das tun, möchte ich die Editoren eurer Zeitschriften bitten, einige weitere "Oldtimer" zu der Liste aus meiner gestrigen e-mail hinzuzufügen:

Bitte fügt Alan Furmann und John Peters in die Liste zwischen George Perry und Jay McKnight ein. Soweit ich mich erinnere, haben sich beide als Oldtimer qualifiziert, denn sie waren schon hier, als ich das erste mal an der SVFIG teilnahm.

Tschüss, *Henry*

P.S.: Sagt mir bitte, ob ich richtigerweise den goldenen Swap-Drachen auf Chuck Moores Hemd sehe. Er sagte mir, daß er ihn bekommen hätte. Wenn mich meine Augen nicht täuschen, sieht man ihn auf der linken Seite seines Hemdes (Ich habe meinen Anstecker auf der rechten Seite getragen).

Anmerk.: Ja, Henry, Charles Moore hat einen goldenen SWAP von der Forthgesellschaft bekommen, ebenso wie Tom Zimmer, Leo Brodie und Elizabeth Rather.

fep

Stehend: John Hall, Bob Nash, Henrik Thurfjell, Dave Jaffe, Chuck Moore, Doug Hammed, John Carpenter, Herman Griffin, Henry Vinerts, Alan Furman, Bob Smith, Dudley Ackerman, Glen Haydon, Byron Nilsen

Knieend: Kevin Appert, Jay McKnight, Ken Morley, John Rible, Andy Korsak, CH Ting, Paul Clifford, Robert Patton, Charley Shattuck

Erste Reihe: John Peters, Barry Cole, Icarus Sparry, George Perry, Jeremy Wade, Jon Mayo, Michael Montvelishsky, Jeff Fox

Windows-Programmierung mit SwiftForth und MySQL

Teil 2

Stefan Schmiedl
<s@xss.de>

Was bisher geschah ...

Neben der Anbindung der libmysql.dll-Bibliothek gibt es bequeme Worte für den Einsatz parametrisierter Strings. In diesem Teil wird eine einfache Batch-Applikation zur Übernahme von Bestandsdaten aus einem COBOL-System beschrieben. Ein paar Zeilen sind auch meiner Entwicklungsumgebung für dieses Projekt gewidmet.
Doch zunächst einmal werfen wir einen Blick auf die **Hausaufgabe** vom letzten Mal: Das wohl größte Problem bei der Implementierung parametrisierter Strings ist die fehlende Kontrolle, ob der Zielbereich ausreicht. Dazu bringe ich im Package folgende Änderungen an:

```
\ -- pstr.f    sws 04-09-05
variable (PEND)

: @SRC++  ( src dest -- src' dest c )
  swap dup>r c@ r> 1+ -rot ;

: (GO?)  ( src dest - src' dest c b )
  @src++ dup 0    <>
  over (param) c@ <>  and
  third (pend) @  <   and ;

: (COPY)  ( src dest -- src' dest' )
  begin
    (go?)  while  over c! 1+
  repeat  drop ;

: [PSTR  ( src dest n -- )
  over + 1- (pend) !
  dup (pstr) !  (pointers) 2!
  (next-block) ;

: >>  ( prm -- )
  (pointers) @ (copy) (pointers) !
  drop  (next-block) ;
```

Bei der Initialisierung wird der Zeiger `(pend)` auf das Ende des Zielbereichs gesetzt, was in der der Übersichtlichkeit halber ausgegliederten Funktion `(go?)` überprüft wird. Auch während der Ersetzung wird die Überprüfung fortgeführt, wodurch der ähnliche Code in >> nun vollends durch den Aufruf von `(copy)` ersetzt werden kann.

COBOL-Datensätze

In dieser Folge beschreibe ich in Auszügen den Aufbau einer selbstständig lauffähigen (Batch-)Applikation, die Daten aus der in COBOL geschriebenen Kunden- und Auftragsverwaltung in die MySQL-Datenbank übernimmt. Eine kurze Ein-

weisung durch den Programmierer, der zufälligerweise im übernächsten Dorf wohnt (der Kunde ist ca. 70 km entfernt), erleicherte das Einlesen der vorhandenen DAT-Dateien ungemein.

Der Quellcode für die Applikation beginnt mit etwas Verwaltungskram:

```
\ -- pdat.f   sws 04-09-05
empty
throw#
  s" Dateizugriffsfehler"
    >throw enum IOR-FILEACCESS
  s" Speicherfehler"
    >throw enum IOR-NOMEMORY
  s" Datensatzlänge"
    >throw enum IOR-NOCOUNT
  s" Formatfehler"
    >throw enum IOR-DATAFORMAT
to throw#
```

SwiftForth verwendet `>throw` dazu, um einer Fehlernummer einen Text zuzuordnen. Der Wert `throw#` enthält die zuletzt definierte Fehlernummer.

Die Eingabedateien beginnen mit einem 128 Zeichen großen Block, der für mich uninteressant ist. Anschließend folgen die Daten in einem "Fast-nur-Text"-Format mit Spalten fester Breite, das ziemlich einfach eingelesen werden kann. Die maximal mögliche Zeilenlänge bleibt mit 2048 Byte gut handhabbar.

```
: OPEN-DAT  ( c[] n -- buffer fid )
  r/o open-file ior-fileaccess ?throw
  2100 allocate ior-nomemory ?throw
  swap ;

: CLOSE-DAT  ( buffer fid -- )
  close-file drop  free drop ;
```

Das Datenlayout selbst ist auch interessant: Der erste Datensatz einer Datei sieht in Auszügen folgendermaßen aus:

```
0080: 401f 3030 3130 ...    @.0010...
0090: 3030 3036 3230 ...    000620...
00A0: 2000 0000 401f ...     ...@....
```

Zu den 31 Byte Nutzlast kommen noch 2 Byte für interne Markierungen und Längenangabe dazu, das Resultat wird auf das nächste Vielfache von 4 aufgerundet, im Beispiel benötigt der 31 Byte lange Datensatz also 31 + 2 = 33 Byte, so dass der zweite Datensatz nach 36 Byte beginnt.

```
: READ-COUNT  ( buf fid -- n )
  over 2 rot read-file
  ior-nocount ?throw drop
  dup c@ $0F and 8 lshift
  swap 1+ c@  + ;

: PADDED  ( n -- n' )
  5 + $FFFC and 2- ;
```

```
: READ-RECORD  ( buf fid -- n flag )
  2dup read-count padded dup>r swap
  read-file ior-fileaccess ?throw
  dup r> = ;
```

Die Worte `read-count` und `padded` verstecken interne Details des Datenformats, `read-record` liefert als Ergebnis die Zahl der gelesenen Byte und einen Indikator für die Gültigkeit des eingelesenen Datensatzes.
Je nach Datei wird ein eigenes Verfahren für die Übernahme in die Datenbank notwendig, das aber für das Abarbeiten aller Datensätze keinen Einfluss nehmen soll.

```
defer RECORD-PARSER  ( addr n -- )
' 2drop is record-parser
```

Da der einleitende „Verwaltungsblock“ selbst auch dem beschriebenen Schema folgt, kann ich beim Verarbeiten der Datei den ersten Block ignorieren.

```
: READ-LINES  ( buf fid -- )
  locals| fid buf |
  buf fid read-record 2drop
  begin
    buf dup fid read-record
  while            \ -- addr n
    record-parser
  repeat  2drop ;
```

Mit dem Wort `parse-dat` wird dann das Einlesen aus der angegebenen Datei mit einem definierbaren Importeur erledigt.

```
: PARSE-DAT  ( c[] n xt -- )
  is record-parser
  open-dat 2dup
  ['] read-lines catch
  ?dup if
    >r 2drop close-dat r> throw
  else  close-dat  then ;
```

COBOL-Datenfelder

Für den Zugriff auf die Datenfelder des aktuell eingelesenen Buffers definiere ich Zugriffsworte, die direkt aus der im COBOL-System verwendeten Feldbeschreibung übernommen werden.

```
: CF:  ( off sz : <name> -- off+sz )
  create  2dup , ,  +
  does>  ( addr -- c[] n )
  2@ >r + r> ;

0  6 cf: KO-LIEFERSCHEIN
   6 cf: KO-KUNDE
   4 cf: KO-AUSFALL%           \ 2.2
   7 2 + 3 * 6 * cf: KO-WAAGEN \ 7.2
drop
```

Die hier gezeigten Beispiele stammen aus dem Überwachungsprotokoll der Waagen in der Gurkenanlieferung. Für jede der sechs Waagen werden drei Gewichte notiert: das vom Vorgän-

ger verbliebene Restgewicht, das Gewicht der vollständig gefüllten Kisten und das Gewicht der letzten, nur teilweise gefüllte Kiste. Diese Gewichte werden als skalierte Ganzzahlen gespeichert. Um nicht all zu viele Zugriffsworte definieren zu müssen, greife ich auf die Messwerte mehrstufig zu:

```
0 27 cf: KO-WAAGE-1
  27 cf: KO-WAAGE-2 ...
  27 cf: KO-WAAGE-6
drop

0  9 cf: KO-VORGEWICHT      \ 7.2
   9 cf: KO-VOLLE-KISTEN
   9 cf: KO-AKTUELLE-KISTE
drop
```

Eine nette Überraschung (zumindest für mich) war die Art, wie negative Zahlen in der DAT-Datei gespeichert werden. Eine Zahl ist negativ, wenn bei ihrer letzten Ziffer ein Markierungsbit ($40) gesetzt ist. Das führt zu folgenden Definitionen:

```
: SIGNED?  ( c -- c' t/f )
  dup $40 and tuck xor swap ;

create (%%INT)  20 allot

: >(%%INT)  ( c1[] n1 -- c2[] n1 )
  (%%int) swap 2dup 2>r cmove  2r> ;

: DIGIT@END  ( addr n -- changed )
  + 1- dup c@ signed? dup>r
  if  swap c!  else  2drop  then r> ;

: %%INT  ( addr n1 -- n2 )
  >(%%int) 2dup digit@end -rot
  number? 1- ior-dataformat ?throw
  swap  if  negate  then ;
```

Zunächst trennt `signed?` das Vorzeichen von der Zahl, so dass mit `digit@end` ein String mit „normalen“ Ziffern in einem Zwischenspeicher erzeugt werden kann. `%%int` schließlich koordiniert diesen Umwandlungsprozess. Die Umwandlung auf gerade diese Art zu machen, erschien mir im Frühjahr noch „natürlich“, heute bin ich damit allerdings nicht zufrieden.

Hausaufgabe:
Wie wandelt man vorzeichenbehaftete COBOL-Zahlenstrings eleganter in brauchbare Ganzzahlen um?

Von DAT zu MySQL

Die beiden Hilfsdefinitionen $,“ und $,0 sind recht brauchbar für die Einbettung längerer SQL-Statements. `REPLACE INTO` ist eine MySQL-Eigenheit, die aber ungemein praktisch ist: Ist bereits ein Eintrag mit dem verwendeten Primärschlüsselwert vorhanden, werden bei diesem die angegebenen Informationen eingetragen. Gibt es noch keinen solchen Eintrag, wird er hinzugefügt.

```
: $,"  ( -- addr )
  [CHAR] " WORD COUNT R-BUF R>
  LOCALS| buf size src |
  src buf size cmove
  buf here size dup allot cmove ;

: $,0  ( -- )
  0 c, ALIGN ;

create SQL.UPDATE-LIEFERANT
$," replace into lieferant
    (kav, lkurz, matchcode, name1,
    name2, strasse, ort) "
$," values (?, '?', '?', '?',
    '?', '?', '?')"
$,0
```

Für die Übergabe der Daten verwende ich die im ersten Teil entwickelten Komponenten:

```
include pstr
include mysql
0 value conn

: CONNECT  ( -- )
  db-init dup to conn
  z" 192.168.1.2"
  z" user" z" pass" z" db"
  db-connect ;

: HANGUP  ( -- )
  conn db-close  0 TO conn ;
```

Der Zugang zur Datenbank ist nach den geleisteten Vorarbeiten schnell geschaffen, so dass wir uns dem letzten noch fehlenden Glied in der Kette zuwenden können. Auf der vorhergehenden Seite habe ich von einem frei definierbaren Importeur gesprochen, der die Umsetzung der eingelesenen Daten vornehmen soll. Er überträgt die vorhandenen Daten in nullterminierte Strings und füllt damit das Abfrage-Template aus.

```
create $data 100 allot
create $sql 1024 allot

: >ZPAD  ( addr n -- z )
  $data zplace $data ;

: %KUNDE>LIEFERANT  ( addr n -- )
  drop
    dup ku-anlieferer s"    "
    compare 0> if
      conn sql.update-lieferant
      $sql 1024 [sql
        dup ku-kunde >zpad       >>
        dup ku-anlieferer >zpad  >>
        dup ku-matchcode >zpad   >>
        dup ku-name1 >zpad       >>
        dup ku-name2 >zpad       >>
        dup ku-strasse >zpad     >>
        dup ku-ort >zpad         >>
      sql]exec
  then drop ;
```

weiter auf Seite 14

Holländisch ist gar nicht so schwer. Es ähnelt sehr den norddeutschen Sprachgepflogenheiten. Und außerdem ist Forth sowieso international. Neugierig? Werden Sie Förderer der

HCC-Forth-gebruikersgroep.

Für 10 € pro Jahr schicken wir Ihnen 5 oder 6 Hefte unserer Vereinszeitschrift 'Het Vijgeblaadje' zu. Dort können Sie sich über die Aktivitäten unserer Mitglieder, über neue Hard- und Softwareprojekte, über Produkte zu günstigen Bezugspreisen, über Literatur aus unserer Forth-Bibliothek und vieles mehr aus erster Hand unterrichten. Auskünfte erteilt:

Willem Ouwerkerk
Boulevard Heuvelink 126
NL-6828 KW Arnhem
E-Mail: ***w.ouwerkerk@kader.hobby.nl***

Oder überweisen Sie einfach 10 € auf das Konto 525 35 72 der HCC-Forth-gebruikersgroep bei der Postbank Amsterdam. Noch einfacher ist es wahrscheinlich, sich deshalb direkt an unseren Vorsitzenden Willem Ouwerkerk zu wenden.

25 Jahre FIG UK

Am 30. Oktober 2004 hat die FIG UK in Morden, in Südengland, ihre 25. Jahrestagung abgehalten. Einen herzlichen Glückwunsch an dieser Stelle an die Adresse unserer englischen Kollegen sende ich als Redakteur der Vierte Dimension" im Namen aller Leser und Mitglieder der Forthgesellschaft.

Friederich Prinz

Zum Treffen der Forther im Silicon Valley:

ForthDay 2004

http://www.forth.org/svfig/fd2004/photos.html

zeigt eine Reihe von Photos mit Forthern, von denen nicht nur Henry Vinerts immer wieder berichtet.
Möchten Sie C.H. Ting einmal an seinem berühmten Grill sehen? Schauen Sie selbst nach.

fep

Liebes Mitglied,
lieber Abonnent,

denken Sie bitte als Überweiser daran, Ihren Beitrag bis zum 14. März 2005 zu entrichten. Den Überweisungsvordruck haben Sie zusammen mit dieser VD erhalten.

Wenn wir von Ihnen eine Einzugsermächtigung haben, so teilen Sie uns bitte Änderungen der Bankverbindung mit. Es erspart vor allem Ihnen (aber auch uns) Kosten für evtl. Bankretouren. Die Banken schlagen inwischen erheblich zu. Sie ersehen aus dem Beitragsbrief, den Sie zusammen mit dieser VD erhalten haben, was wir gespeichert haben. Ihr Beitrag wird am 14. März 2005 eingezogen.

Herzlichen Gruß, Ihr Forth-Büro

Rolf Schöne

(Englische Forth-Gesellschaft)

Treten Sie unserer Forth-Gruppe bei.
Verschaffen Sie sich Zugang zu unserer umfangreichen Bibliothek.
Sichern Sie sich alle zwei Monate
ein Heft unserer Vereinszeitschrift.
(Auch ältere Hefte erhältlich)
Suchen Sie unsere Webseite auf:
www.users.zetnet.co.uk/aborigine/Forth.htm
Lassen Sie sich unser Neuzugangs-Gratis-Paket geben.
Der Mitgliedsbeitrag beträgt 12 engl. Pfund.
Hierfür bekommen Sie 6 Hefte unserer
Vereinszeitschrift Forthwrite.
Beschleunigte Zustellung (Air Mail)
ins Ausland kostet 20 Pfund.
Körperschaften zahlen 36 Pfund,
erhalten dafür aber viel Werbung.

Wenden Sie sich an:

Dr. Douglas Neale
58 Woodland Way
Morden Surrey
SM4 4DS
Tel.: (44) 181-542-2747
E-Mail: dneale@w58wmorden.demon.co.uk

Fortsetzung von Seite 12

Eine Batch-Applikation

Die Übernahme der gezeigten Daten kann nun kurz und bündig erfolgen, würden wir in der Forth-Konsole bleiben, wären wir fertig:

```
: IMPORT
  connect
  s" daten\kunden.dat"
    ['] %kunde>lieferant parse-dat
  hangup ;
```

Es ist kein großer Aufwand, mit SwiftForth aus dem beschriebenen System eine eigenständige Applikation zu erstellen. Damit die Benutzer zumindest eine minimale Rückkopplung erhalten, garniere ich den Importvorgang noch mit einer Statusmeldung am Ende und referenziere dieses Wort über den dafür in SwiftForth vorgesehenen Vektor `'main`. Schließlich erzeugt `program` eine Win32-konforme exe-Datei mit dem angegebenen Namen.

```
: GO
  ['] import catch if
    0 z" Fehler beim Datenimport!"
    ebox 1
  else
    0 z" Datenübernahme ok."
    ebox 0
  then ExitProcess ;
' go 'main ! program IMPORT.EXE
```

Werkzeug

In meinem Werkzeugkasten für dieses Projekt befanden sich nur wenige Hilfsmittel: SwiftForth, gvim und CVS-NT. Letzteres (Projektgedächtnis und Sicherheitsnetz) musste nicht weiter angepasst werden.
Zunächst einmal stört natürlich, dass gvim glaubt, bei Dateien mit Erweiterung .f handele es sich um Fortran, entsprechend unpassend ist die Syntaxfärbung des Quellcodes. Solch unpassende Vorgaben schalte ich ab.

```
filetype off
syntax off
set iskeyword=33-255
map <F12> ve~e
```

Für (mehr oder weniger wild aussehende) Forth-Worte lege ich mit `iskeyword` fest, dass alle sichtbaren Zeichen wie Buchstaben behandelt werden. Damit kann ich auf einem Wort mit der Taste # zum vorherigen Erscheinen in der aktuellen Datei springen, auch wenn Sonderzeichen in ihm vorkommen. Die Taste F12 schreibt das Wort ab dem Cursor groß, womit ich neu definierte Worte kennzeichne.
Die folgenden Definitionen ermöglichen ein "Zusammenfalten" des Quellcodes, mit dem sich eine strukturierte Übersicht blitzschnell erzeugen lässt.

```
set foldmethod=marker
set foldlevel=0
set foldtext=ForthFolder()
set fillchars=stl:\ ,stlnc:\ ,
    vert:\|,fold:\ ,diff:-
function ForthFolder()
  let l:sec = v:folddashes .
    v:folddashes .
    substitute(getline(v:foldstart+1),
       '^.....', '', '') . '  '
  let l:cnt = '  [' . (v:foldend-
    v:foldstart+1) . ']  '
  let l:fc = 75 - strlen(l:sec) -
    strlen(l:cnt)
  let l:fil = strpart('------------------
----------------------------------------
--------------', 0, l:fc)
  return l:sec . l:fil . l:cnt
endfunction
```

Wie viel Code sichtbar ist, wird über den so genannten Foldlevel gesteuert, der sich mit einigen Abkürzungen bequem einstellen lässt (die seltsame Wahl kommt daher, dass vimoutliner die gleichen Kombinationen einsetzt) :

```
nmap ,,1 :se fdl=0
nmap ,,2 :se fdl=1
nmap ,,0 :se fdl=9
```

Unmittelbar nach dem Öffnen sehe ich im Editor in etwa Folgendes:

```
--Dateiverwaltung  ----------  [35]
--Parsing-Strategie  --------  [24]
--Def. der Datenfelder  -----  [17]
--Verarbeitung  -------------  [29]
--Daten-Ausgabe  ------------  [35]
--Parser-Worte  -------------  [55]
--MySQL+Applikation  --------  [88]
```

Schließlich verwende ich noch eine minimale Syntaxfärbung: Kommentare (\ bis Zeilenende und in () eingeschlossen) sind blau, alles übrige schwarz.

```
au BufRead *.f syntax clear
au BufRead *.f syntax case ignore
au BufRead *.f syntax match Comment
   /^\\ .*$/
au BufRead *.f syntax match Comment
   / \\ .*$/
au BufRead *.f syntax match Comment
   / ( .\{-})/
```

Natürlich konnte ich auch SwiftForth nicht unangetastet lassen, denn obwohl eine 500 kB große Datei für Windows-Verhältnisse wohl zu den Fliegengewichten zählt, ist es doch weit entfernt von dem, was eigentlich notwendig wäre.
Blättern im Quellcode hat mich auf die Datei src\ide\win32\hi.f gebracht, in der die IDE auf den ca. 50 KB großen SwiftKernel aufgesetzt wird. Durch Streichen der IDE-relevanten Teile konnte ich die Größe einer Applikation halbieren, so dass

sich die Größe der in der Konservenfirma eingesetzten Applikätiönchen nur noch zwischen 250 kB und 280 kB bewegt. Ein kompletter Aufbau aller acht eingesetzten Applikationen nimmt (beginnend vom Aufbau eines schlanken Systems aus dem Kernel und der modifizierten hi.f bis zur Verpackung in ein ZIP-Archiv) auf meinem Notebook etwa sieben Sekunden in Anspruch, wovon 5 auf die Kompression entfallen und der Löwenanteil der übrigen Zeit vermutlich für das Öffnen und Schließen eines Forth-Konsolenfensters verwendet wird, das ich noch nicht unterbinden konnte.

Wie geht's weiter?

Im letzten Teil beschreibe ich eine ziemlich minimale GUI-Applikation, die (vom Kunden) als ungemein nützlich empfunden wird, obwohl sie oberflächlich nur aus einem Eingabefeld und vier Schaltflächen besteht.

Stefan Schmiedl

Irrtümer der Schulbücher

Adolf Krüger hat es entdeckt, *Michael Kalus* hat es der VD geschickt und ich meine, daß ich meinen Spaß an der nachfolgenden Adresse mit Ihnen teilen sollte.

http://www.amasci.com/miscon/miscon4.html

William J. Beaty klärt die geneigten Leser dort über "Misconcepts" in K-6 Schulklassen auf. Der K-6 Bereich umfaßt in den USA die ersten 6 bis 7 Schuljahre (Elementarschulbereich; 5 bis 11 Jahre). Was Hänschen nicht lernt, lernt Hans nimmermehr, heißt es in einer Volksweisheit. Umgekehrt läßt sich vermuten, daß Hans (Fritz natürlich auch nicht) niemals wirklich „ablegen" wird, was man ihm in den ersten Schuljahren eingetrichtert hat.

Darum versucht Beaty, Elektroingenieur und Lehrer, mit Erklärungsversuchen aufzuräumen, die mit groben Fehlern, unzulässigen Vereinfachungen, falschen und bisher unkorrigierten Voraussetzungen oder einfach mit dem offensichtlichen Unwissen der Lehrer dazu beitragen, Schüler dieser Lernstufe "auf Lebenszeit" zu verwirren.

Wissenschaftler arbeiten mit wissenschaftlichen Methoden? Nicht ganz, meint Beaty. Der Himmel scheint aufgrund komplizierter, physikalischer Zusammenhänge eine blaue Farbe zu haben? Das ist falsch, sagt Beaty, und erklärt gleich auch noch, warum Wasser blau ist, und daß Wasser und Luft in Wahrheit nicht durchsichtig, sondern "nur" durchscheinend sind. Der Erklärung für Gleitreibung als Folge rauher Oberflächen setzt Beaty jüngere Erkenntnisse aus der Chemie entgegen, und das Wissen darum, daß der Energieverlust durch das „Aufheben" von Lasten aus einem „Friktionstal" durch Energiegewinn beim anschließenden „Hineinfallen" der Last in ein neues Tal kompensiert wird.

Viel Spaß mit dieser englischsprachigen Seite!

fep

Grüße von Trägern des goldenen SWAP

Die Forthgesellschaft hat in den USA selbstverständlich auch Elizabeth Rather, Charles Moore, Leo Brodie und, last not least, Tom Zimmer mit dem golden SWAP und jeweils einer kleinen, persönlichen Danksagung für ihre forthigen Leistungen geehrt.

Leo Brodie und Tom Zimmer haben sich dafür per Mail bedankt und darum gebeten, den Mitgliedern der Forthgesellschaft ihre Grüße auszurichten.

Leo Brodie (13.Okt. 2004):

I received it yesterday!

It is a lovely commemoration. I am touched by the kind words in the letter.
Thank you so much for thinking of me. Please pass these thanks on to your group.
Warmly,
Leo Brodie

+—+—+—+—+—+—+—+

Tom Zimmer (03.Okt. 2004):

Thank you PhiHo, and everyone,
It is always a joy for me to see Forth continuing to be used, and to evolve, and recognition doesn't hurt either.
I have been tempted to describe this award as my "Life Achievement Award", which I suppose in some ways it is, at least with respect to my life using Forth. Of course there is, as some of you know, much more to my life than Forth. True life achievement comes to me not by my own effort, but by the One who is greater working through me. Ultimately, being a servant, provides a glimpse of the greater rewards to come.
Philosophical with a purpose,
Tom Zimmer

Here below is the email response I sent to the FG group, along with their notification;

...
Well, this is a surprise, I appreciate the honor, and it is good to hear that Forth is still going strong. Please convey my appreciation to the FG members, and tell them I wish them another successful 20 years.
I have not used Forth professionally in several years, but I still regularly use my WinView editor, written with Win32Forth. Support for Win32Forth has now been taken over by a group on yahoo groups at;

http://www.forth.org/svfig/Win32Forth/yahoo.html

They have been evolving Win32Forth in many interesting and useful ways.
Best Regards,
Tom Zimmer

Tom Zimmer (13.Okt. 2004)
...
I wanted to drop you a note to let you know that the package just arrived in good condition. I am truly honored and blessed to receive this award. I am an avid motorcyclist, so I will wear the pin prominently on my leather "biker" vest. Thank you and "Forthgesellschaft" members very much,
Best regards,
Tom Zimmer

Sicheres Mailen

(auch mit Windows, Mozilla und GnuPG)

Bernd Paysan

<bernd.paysan@gmx.de>

Kein Scherz: Die Infrastruktur zur flächendeckenden, verdachtsunabhängigen Überwachung von E-Mails wird in Deutschland gerade aufgebaut. Und wenn die Infrastruktur mal da ist, dann wird sie erfahrungsgemäß auch früher oder später genutzt. Macht aber nix, kann man ja verschlüsseln.

Sichere Mail-Kommunikation dient dabei nicht nur dem Zweck, Inhalte vor ungebetenen Gästen zu sichern, sondern schafft auch mehr Vertrauen:

* Signierte Mails sind vertrauenswürdig - zum Überprüfen des Schlüssels braucht man ohnehin die ganze Kryptographie-Infrastruktur.
* Verschlüsselte Mails werden nicht versehentlich als SPAM aussortiert und bis Spammer verschlüsseln, wird wohl noch einige Zeit vergehen.
* SPAM und Würmer kommen (bislang) nicht signiert und verschlüsselt an.

Mit Linux ist die Infrastruktur auch in der Regel vorhanden, MacOS-X und Windows-Benutzer müssen noch etwas Hand anlegen (Sicheres Internet mit Windows ist halt etwas schwerer). Das geht trotzdem recht einfach, zumindest wenn man den Mozilla oder Thunderbird verwendet. Der Einfachheit beschränke ich mich hier auf Mozilla 1.7.5/Thunderbird 1.0 und passendes EnigMail 0.89.6 auf Windows.

GnuPG installieren

* Erst mal den GnuPG für Windows <**ftp://ftp.gnupg.org/gcrypt/binary/gnupg-w32cli-1.2.5.zip**> saugen (Zip-Datei).
* Nach **C:\gnupg** auspacken, und in der Registry den Key auf den gnupg-Pfad setzen (siehe Datei **gnupg-w32.reg**).
* Das war's schon.

Enigmail installieren

* Nötig ist mindestens Mozilla <**http://www.mozilla.org/products/mozilla1.x/**> Version 1.6 (besser 1.7.3 oder neuer), bzw. Thunderbird 0.9 (besser 1.0), um die neuste Enigmail-Version zu benutzen.
* Dann einfach auf die Enigmail-download page <**http://enigmail.mozdev.org/download.html**> gehen. Die versucht, OS und Browser-Version automatisch zu erkennen. Da muss man nur etwas nach unten scrollen, zur Download-Tabelle. Für Mozilla reicht es, auf einen der "Install"-Buttons zu klicken (mit der richtigen Mozilla-Version).
* Bei Thunderbird muss man die passende Version aus der mittleren Spalte laden. Im Thunderbird dann unter "Tools/Extensions" auf "Install" klicken, und die eben abgespeicherte xpi-Datei wählen.
* Jetzt einfach das Mail-Programm neu starten. EnigMail muss noch konfiguriert werden. Manchmal braucht's auch einen Reboot, damit Windows wieder einen klaren Kopf hat.

Konfiguration

* Im "Enigmail"-Menü auf "Preferences" klicken.
* Mit dem "Browse..."-Knopf den GPG-Pfad festlegen; dassollte **C:\gnupg\gpg.exe** sein (oder wo man das gpg.exe halt hingepackt hat).
* Die anderen Optionen dürften schon richtig voreingestellt sein.
* Dann im "Enigmail"-Menü auf "OpenPGP Key Management" klicken, und dort unter dem "Key"-Menü auf "Generate Key" klicken.
* Die Konfiguration ist ausgesprochen einfach: In der Dialogbox kann man direkt den Mail-Account auswählen, für den man den Key erzeugen will.
* Also braucht man nur noch eine *passphrase*. Da muss man Vorsicht walten lassen! Diese Phrase sollte kompliziert genug sein, damit sie niemand einfach so errät, aber einfach genug, damit man sie nicht vergisst.
* Zum Schluss noch auf "Generate Key" klicken.

Benutzen

Den öffentlichen Schlüssel sollte man natürlich bekannt geben. Als Keyserver in Deutschland kommt z.B.

hkp://random.sks.keyserver.penguin.de

in Frage. Dieser Keyserver ist bei EnigMail schon vorkonfiguriert. Dorthin sollte man seinen Schlüssel schicken. Die Schlüssel anderer Leute kann man dann ebenfalls über diesen Keyserver bekommen (so sie ihren Schlüssel hochgestellt haben).
Viele Keyserver sind miteinander verbunden, und tauschen ihre Schlüssel aus - über kurz oder lang tauchen also auch woanders hochgestellte Keys auf.
So ein Schlüssel hat eine 32-Bit-ID, und einen Fingerprint (30

Byte). Natürlich kann man auch über Name und E-Mail nach einem Key suchen – wenn man nach einem Michael Maier oder Meier sucht, kein aussichtsreiches Verfahren.

Wenn man die Schlüssel beisammen hat, kann man endlich loslegen, und verschlüsselte/signierte E-Mails verschicken und empfangen.

Technischer Hintergrund

Das Verfahren, das PGP und GnuPG verwenden, ist ein sogenanntes "Public Key"-Verfahren.
<**http://en.wikipedia.org/wiki/Public_key**>
Eine solche Verschlüsselung ist "asymmetrisch", man braucht also zwei unterschiedliche Schlüssel. Das bekannteste Verfahren ist das RSA-Verfahren,
<**http://de.wikipedia.org/wiki/RSA-Kryptosystem**>
nach den Erfindern Rivest, Shamir und Adleman benannt.

Dabei wählt Alice (die allgegenwärtige Person "A" in der Kryptographie) zwei lange, etwa gleich große Primzahlen p und q. Diese beiden Primzahlen geben als Produkt N den Zahlenraum an, in dem gerechnet wird (immer mod N). Die Operation, die man zum Ver- und Entschlüsseln verwendet, ist die Potenz. Man kennt die Anzahl teilerfremder Zahlen von N, es ist phi(N) =(p-1)*(q-1). Diese Zahl ist zunächt mal nur eine nützliche Hilfszahl zur Berechnung der eigentlichen Schlüssel. Nun wählt man eine beliebige teilerfremde Zahl e zu phi(N). Den anderen Schlüssel bestimmt man durch Lösen der Gleichung e*d mod phi(N) = 1. N und e kann Alice jetzt an Bob (die Person "B") weitergeben, sie stellen den öffentlichen Schlüssel dar. d behält Alice für sich, es ist ihr geheimer Schlüssel.

Wenn Bob also eine geheime Botschaft an Alice schicken will, dann potenziert er seine Botschaft K zu C=K^e mod N. Alice berechnet C^d mod N, berechnet also K^e*d mod N (und wie der höhere Mathematiker unschwer erkennen kann, ist das dasselbe wie K^1 mod N, also wieder K).

Für größere Datenmengen ist das RSA-Verfahren zu aufwendig. Schließlich sind d, e und N ziemlich große Zahlen. Auch die Tricks mit dem Zerlegen von d und e in Summen kleiner Zahlen und kleine Faktoren hilft nur begrenzt weiter. Deshalb wird tatsächlich nur ein kurzer Schlüssel für ein effizientes symmetrisches Verfahren wie AES
<**http://de.wikipedia.org/wiki/Advanced_Encryption_Standard**>
verschlüsselt. Dieser Schlüssel wird nur einmal benutzt, kann aber gleich mehrmals (mit mehreren Schlüsseln) verschlüsselt in derselben Mail untergebracht werden. Damit wächst die Datenmenge dann auch für mehrere Empfänger nicht wesentlich an.

Dieses asymmetrische Verfahren kann Alice auch verwenden, um Bob zu zeigen, dass die Antwort auf seine Mail wirklich von ihr kommt. Alice verschlüsselt dazu ihre Botschaft K mit ihrem geheimen Schlüssel, berechnet also C=K^d mod N. Diese Botschaft kann jeder (auch Bob) mit dem öffentlichen Schlüssel entschlüsseln, indem er C^e mod N berechnet (auch hier wieder K^d*e mod N = K^1 mod N). Eine sinnvolle Botschaft kann sich also nur ergeben, wenn es tatsächlich der geheime Schlüssel von Alice war. Bei dieser Signatur wird allerdings normalerweise nicht die Nachricht selbst verschlüsselt, sondern nur ein sicherer Hash, etwa mit dem SHA-Algorithmus
<**http://de.wikipedia.org/wiki/SHA1**>
berechnet.

Linux

Unter Linux empfehle ich den Kmail. Der hat die nötigen Plug-Ins schon eingebaut. Der kgpg verwaltet Schlüssel auch sehr komfortabel. In die **.xsession** muss man ggf. noch den Aufruf für den **gpg-agent** einbauen. Bei mir sieht das so aus:

eval $(gpg-agent --daemon --keep-display --default-cache-ttl 1800 --allow-mark-trusted)

Das mit der **ttl** ist Geschmackssache - länger ist zwar komfortabler, aber eben auch unsicher.

MacOS-X

/von **Ulli Hoffmann**/

Für Apple-Mail
<**http://www.apple.com/de/macosx/features/mail/**>
gibt's auch ein Plugin namens GPGMail
<**http://www.sente.ch/software/GPGMail/English.lproj/GPGMail.html**>.
So wird's installiert:

1. GPGMail erwartet ein bereits installiertes gpg
<**http://macgpg.sourceforge.net/**>.
Das muss zuerst installiert werden. Also:
* Lade die aktuelle Version des GNU Privacy Guards
<**http://macgpg.sourceforge.net/**>
für Deine Mac OS X Version.
* Aktiviere das GnuPG Mac OS X Festplatten-Image und lass den .mpkg Installer laufen.

2. GPGMail hat keine eigene Schlüsselverwaltung. Um also die Schlüssel nicht direkt per gpg auf der Kommandozeile verwalten zu müssen, bietet es sich an, GPGKeys
<**http://macgpg.sourceforge.net**/>
zu installieren:
* Lade die aktuelle Version von GPGKeys
<**http://macgpg.sourceforge.net/**>.
* Expandiere die **.tar.gz** Datei und ziehe die GPGKeys-Datei in Deinen Applikations-Ordner.
* Zum Generieren Deines neuen Schlüssels starte GPGKeys. Aus dem "Schlüssel"-Menü, wähle "Erstellen...". Es öffnet sich ein Terminal-Fenster. Dort solltest Du die Voreinstellungen belassen, es sei denn Du weist genau, was Du tust. Wenn Du gefragt wirst, gibst Du Deinen echten Namen, Deine Email-Adresse und einen Kommentar (Kurznamen) an. Wenn der Schlüssel generiert wurde, kannst Du das Terminalfenster

schließen.

3. Jetzt kommt endlich GPGMail dran:
* Lade das GPGMail
  <**http://www.sente.ch/software/GPGMail/English.lproj/GPGMail.html#Download**>
  plug-in von Sen:te für Apple-Mail.
* Aktiviere das Festplatten-Image und starte das "Install GPGMail" Skript.

4. Starte Mail neu. Alles sollte jetzt funktionieren. Wenn Du eine neue Nachricht erstellst, siehst Du eine Check-Box, die Dich Deinen Schlüssel wählen läßt und Dir erlaubt, die Nachricht digital zu unterschreiben bzw. zu verschlüsseln. Wenn Du eine signierte Nachricht erhälst, bist Du in der Lage sie zu verifizieren. Für Dich verschlüsselte Nachrichten kannst Du entschlüsseln.

Anonymes Surfen

Friederich Prinz

(Friederich.Prinz@t-online.de)
(VD@forth-ev.de)

Bernd Paysans vehement vorgetragene Aufforderung an die Direktoren, den Editor der Vereinszeitschrift und das Forthbüro, den bürgerrechtsbedrohenden Tatsachen ins Auge zu sehen, und endlich auf die vorhandenen Möglichkeiten verschlüsselter eMail zuzugreifen, hat zunächst eine Kontroverse innerhalb des gerade aufgezeigten Kreises herbeigeführt.

Nichts zu verbergen zu haben ist ein angenehmes Ruhekissen. Das gilt zumindest solange, wie niemand die vorhandenen Möglichkeiten, die „der Staat“ sich aktuell schafft, gegen mich verwendet. Leider fehlt mir die Vorstellungskraft, um mir auszumalen, was sich aus meinen eMails tatsächlich herauslesen oder aber böswillig hinein interpretieren ließe. Mein Vertrauen in die aktuell handelnden Personen der Politik, ist allerdings schon sehr begrenzt. Über meine Ansichten bezüglich der persönlichen Integrität vormals grüner, heute vorgeblich roter und tatsächlich rot-grün koalierender Minister bin ich gerne bereit via Mail zu korrespondieren, gerne auch unverschlüsselt. Unverschlüsselt deshalb, weil es hier nicht um eine Paranoia geht. Es ist nach meiner persönlichen Einschätzung auch eher unwahrscheinlich, daß sich Herr Schily oder die von ihm befehligten Organe ausgerechnet für mich und meinen elektronischen Verkehr interessieren. Es geht schlicht um die Wahrung eines Bürgerrechtes.

„Wer seine Bürgerrechte nicht wahrnimmt, hat sie gar nicht verdient“, schreibt Bernd Paysan in einer ebenfalls unverschlüsselten eMail. Und dem ist eigentlich nichts hinzu zu fügen; es sei denn der Hinweis, daß selbstverständlich nicht nur eMail von der Beugung des Grundgesetzes betroffen ist. [GG BRD, Art. 10, Abs. 1 – „Das Briefgeheimnis sowie das Post- und Fernmeldegeheimnis sind unverletzlich.“]

Die *Telekommunikations-Überwachungsverordnung* gilt ohne Ausnahme für ALLE Telekommunikationseinrichtungen und Verfahren, (außer für solche, von denen man vermuten muß, daß sie von jenen betrieben werden, die Herr Schily bespitzeln zu wollen vorgibt). Das heißt, daß z.B. Ihr Internetprovider festhalten muß, wo, wann und wie Sie sich auf welcher Web-Page herumtreiben. Informieren Sie sich vielleicht heimlich über das Parteiprogramm der CDU? Haben Sie bei der “Wahlalternative Arbeit und soziale Gerechtigkeit“ nach einer Orientierung gesucht? Oder waren Sie das, der letzte Woche mit Google nach der Liste gesucht hat, auf der alle (gegenleistungsfreien) Nebeneinkünfte aller PolitikerInnen in den Parlamenten dieser Republik verzeichnet sein sollen?

Möchten Sie zulassen, daß diese und andere Informationen über Sie von beliebigen Sachbearbeitern in beliebigen Behörden aufgegriffen und „bearbeitet“ werden?

Das möchten Sie nicht?

Dann schützen Sie sich eben auch dagegen. Möglichkeiten gibt es genug.

Kommerzielle Software wie „Steganos“ bietet teilweise solche Möglichkeiten des Selbstschutzes. Steganos ermöglicht es, sich in weiten Bereichen „anonym“ im Internet zu bewegen. Web-Seiten, die Sie besuchen, können nicht auf Ihre tatsächliche URL zurückgreifen. Selbst Ihr Provider bleibt unerkannt.

Varianten von Steganos, wie der AisAliveProxyServer, ein Shareware-Produkt, leisten Gleiches, benötigen aber ein wenig mehr Einarbeitungszeit des Nutzers.

Wenn es nur darum geht, dort unerkannt zu bleiben, wo man sich tummelt, dann reicht auch schon ein kleiner Proxy-Server wie der CCProxy von YoungzSoft. Der bekommt die Adresse eines anonym arbeitenden „echten“ Proxys mitgeteilt. Und der heimische Browser nutzt den CCProxy als Filter zum Web. Wie man das alles macht, erkläre ich gerne per eMail, oder, wenn ausreichende Nachfrage besteht, in der nächsten Ausgabe der VD. Wenn Sie bis dahin selbst probieren wollen, “unerkennbar“ zu werden, dann fragen Sie sich vielleicht, wie Sie an die Adresse eines wirklich anonym arbeitenden Proxys kommen. Schauen Sie einfach einmal bei

<**http://www.stayinvisible.com**>

unter Public-Proxy-Servers nach. Probieren Sie die mit “high anonymity“ gekennzeichneten Server einfach aus.

Und seien Sie versichert, daß Sie damit nichts Ungesetzliches tun! Ihr Recht darauf, Ihre Rechte zu schützen, könnte man Ihnen allenfalls per Verordnung nehmen, wenn das Grundgesetz

nun auch noch einen Artikel enthielte, der in etwa hieße: „Alle Menschen haben das Recht, für ihre Menschen- und Bürgerrechte uneingeschränkt einzutreten und diese Rechte gegen jedermann und jegliche Institution, insbesondere gegenüber jeder politischen Partei zu verteidigen.“ Das könnte gerade heute handelnde, politische Personen zu einer Verordnung verleiten, genau dieses Recht wegen eines inneren Notstandes auszusetzen.

Ist es nicht geradezu wahnwitzig, daß ein elementares Recht wie der Schutz der eigenen Bürgerrechte vor allem dadurch selbst geschützt wird, daß dieses Recht eben nicht eindeutig und verbindlich niedergeschrieben wurde?

Ähnliches mögen sich die Entwickler des JAP gedacht haben, der, wie alle MIXe, weit über das bloße Unkenntlichmachen der IP hinausgeht. Mit Details über den JAP (Java AN.ON Proxy) können Sie sich detailliert an der folgenden Adresse versorgen:

http://anon.inf.tu-dresden.de/

Dort finden Sie nicht nur den sehr einprägsamen Satz “**Anonymity is not a crime**“, sondern auch einen tatsächlich sehr weitgehenden Schutz Ihrer Privatsphäre im Internet.

Der JAP ist ein Proxy-Server, der bereits bei Ihnen zuhause mit dem Verschlüsseln Ihrer Daten beginnt. JAP nimmt Kontakt mit dem MIX der TU Dresden, oder der Universität Regensburg auf, und sorgt dafür, daß alle Ihre Daten auf dieser Verbindung erst einmal nicht für jedermann lesbar sind. Die MIXe sorgen anschließend dafür, daß externe Beobachter, wie zum Beispiel die Mitarbeiter des Herrn Schily, vielleicht noch feststellen können, daß ausgerechnet SIE einen MIX nutzen, aber das niemand mehr sagen kann, wohin Ihre Surftouren Sie führen und welche Dateien Sie woher über das Web-FTP herunterladen!

Dieser Dienst der Hochschule, der zwar schon recht stabil arbeitet, aber immer noch in der Entwicklung ist, ist zur Zeit noch kostenfrei zu haben, soll aber im kommenden Jahr zu einem “Outstand“ werden und seine Mannen selbst ernähren. Bis dahin ist die öffentliche Finanzierung dieses technisch überaus interessanten und aus Datenschutzgründen ungemein wertvollen Projektes noch gesichert.

Was den Datenschutz anbelangt, berührt es schon merkwürdig, daß ausgerechnet „schwarze“ Landesregierungen (Sachsen und Bayern) der „roten“ Bundesregierung mit Hinweis auf die Bürgerrechte auf die gierigen Finger hauen müssen. Noch intensiver berührt die Information, daß es auch bezüglich des Dresdener JAPs mit dem Datenschutz nicht ganz so weit her ist, wie es auf den Internetseiten der TU propagiert wird. Mindestens in einem belegten Fall hat der JAP bereits einem Begehren des Bundesnachrichtendienstes nachgegeben und Daten eines Nutzers mitgeschrieben und weiter gegeben.

Die hierfür sinnvollerweise anzunehmenden Gründe (Angst vor dem Verlust des Arbeits- oder Studienplatzes, Angst vor Sanktionen mit Hilfe des Beamtenrechts, Angst um den Fortbestand des Projektes) sind nachvollziehbar und meines Erachtens sogar akzeptabel. Das Vertrauen in den JAP, bzw. in den Dresdener MIX, fördern sie indes nicht. Denn selbst wenn der Betreiber des MIXes (die TU Dresden) die mitgeschriebenen Daten von Rechts wegen herausgeben mußte, hätte er zuvor durch eine Information an die damaligen Nutzer und durch anschließendes Abschalten des MIXes bereits das Mitschreiben von Daten verhindern, bzw. unterlassen können - und dies ohne eine größere Sorge als die, ob die Testnutzer “bei der Stange bleiben“.

Immerhin gibt es einen weiteren MIX, das bereits erwähnte System der Universität Regensburg. Dieses mag in interessierten Kreisen schon deshalb größeres Vertrauen potentieller Nutzer gewinnen, weil es mit dem CCC in Hamburg (ChaosComputerClub) “mixt“. Zwar sind die Mitglieder des CCC heute durchaus auch gereifte Persönlichkeiten, aber ich persönlich vermute nach wie vor, daß gerade diese Gruppierung mit Argusaugen darüber wacht, was ihr “Partner“ alles so treibt. Und weil bei diesem MIX auch die Berliner Humboldt-Universität mitmischt, bin ich zuversichtlich, daß meine Versuche, meine Bewegungen im Netz zu anonymisieren, auf dem Weg über diesen MIX erfolgreich bleiben.

Bleibt die Frage, wie sich mit dem MIX in Regensburg Kontakt aufnehmen läßt. Das geht am einfachsten über den JAP. Damit habe ich aber nach wie vor das Vertrauensproblem gegenüber dem vorgeschalteten Infoserver der TU-Dresden, der potentiell meine Daten mitschreibt, bevor er sie nach Regensburg weiterleitet. Vielleicht finden Sie durch Herumspielen mit dem JAP einfach heraus, auf welchen Ports der anonymizer.ccc.de (80.237.206.62), der unmittelbar für die Universität Regensburg zu arbeiten scheint, Ihren Anfragen gegenüber offen ist. Den MIX in Dresden erreichen Sie übrigens auch ohne den Infoservice auf dem **Port 443** (SSL Übertragungen) unter der Adresse **mix.inf.tu-dresden.de**. Damit ist die Strecke zwischen Ihnen und dem MIX nach wie vor “offen“, aber wenn Sie nicht bereits andere Wege nutzen, sind Sie damit in jedem Fall besser dran als in diesem Augenblick.

In diesem Sinne wünsche ich Ihnen und mir, daß wir uns im Netz bei der Suche nach interessanten Techniken und Möglichkeiten eben nicht sehen.

F.....P....

Charles Moore‘s Grüße an die FG

I received your Golden Swap Pin. Thank you very much for the honor it represents. Let's look forward to 20 more years of Forth.

My congratulations to whoever designed the pin. It's a very clever and appropriate symbol.

RSA --- eine Modellimplementierung

Bernd Paysan

1. Januar 2005

Zusammenfassung

Der RSA-Algorithmus von Rivest, Shamir und Adleman ist wohl der bekannteste asymmetrische "Public-Key"-Algorithmus. Die hier beschriebene Forth-Code behandelt alle Teilprobleme, die sich bei der Schlüsselgenerierung und der eigentlichen Codierung ergeben. Für ernsthafte Zwecke ist das Programm nicht geeignet, dazu ist die maximale Schlüssellänge zu kurz (auf 64-Bit-Systemen 64 Bit, auf normalen PCs nur 32 Bit). Ein brauchbarer Schlüssel sollte 1024 Bits lang sein. Der Code ist dennoch so aufgebaut, dass auch eine effiziente Implementierung mit Bignums möglich ist.

Einleitung

Die Lauscher sind überall --- ob es die "üblichen Verdächtigen" vom CCC sind, die sich den einen oder anderen Scherz erlauben, und Webseiten verunstalten, Schlapphüte aus Pullach auf Terroristenjagd oder Spyware-Autoren, die nach Kreditkartennummern auf ungesicherten Windows-Rechnern suchen: Angeblich haben 80% der PC-Benutzer schon mal unfreiwillig ein Spywareprogramm auf ihrem PC gehabt. Es macht schon Sinn, Daten zu verschlüsseln. Das RSA-Kryptosystem ist ein "asymmetrisches" Kryptosystem, d.h. es verwendet verschiedene Schlüssel zum Ver- und Entschlüsseln. Benannt ist das 1977 entwickelte System nach seinen Erfindern Ronald L. Rivest, Adi Shamir und Leonard Adleman.

Ich werde die Kernelemente dieses Systems Schritt für Schritt präsentieren, gebe hier aber gleich eine Warnung: Ein Verschlüsselungsprogramm braucht mehr als nur den Kernalgorithmus. Private Schlüssel müssen sicher aufbewahrt werden, öffentliche Schlüssel komfortabel verteilt werden. Da man RSA aus Performance-Gründen nur verwendet, um sichere Hashs und symmetrische Schlüssel zu verschlüsseln, fehlt für den eigentlichen verschlüsselten Datenaustausch ein symmetrisches Verschlüsselungsverfahren (z.B. AES), und ein sicherer Hash (z.B. SHA1).

Und natürlich fehlt noch eine Möglichkeit, genügend lange Schlüssel zu verarbeiten. Dazu braucht man Bignums, also Zahlen, die viele Zellen lang sind. Erst ab 1024 Bit wird ein Schlüssel als brauchbar lang angesehen. Die Schlüssel, die dieses Programm verarbeiten kann, sind nur eine Zelle lang --- auf einem normalen PC also 32 Bit, auf Workstations oder Athlon64-Rechnern 64 Bit.

Achtung: Kryptographie ist höhere Mathematik. Dieser Text enthält größere Mengen davon.

1 Schlüssel generieren

Der Algorithmus ist nicht schwierig (Quelle: Wikipedia):

1. Wähle zufällig und stochastisch unabhängig zwei Primzahlen $p \neq q$, die etwa gleich lang sein sollten, und berechne deren Produkt $\mathcal{N} = p \cdot q$.

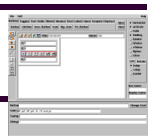
Wir brauchen also erst mal einen Zufallsgenerator. Der hier folgende generiert zwar nur Pseudo-Zufallszahlen, aber er holt sich zumindest über die Uhrzeit etwas Entropie. In der Praxis wäre so etwas ziemlich tödlich, weil genau diese Uhrzeit im Schlüssel mitgegeben wird. Kryptographisch harte Zufallsgeneratoren gewinnen ihre Entropie aus zufälligeren Daten als der Uhrzeit. Das Prinzip wird trotzdem verdeutlicht:

```
\ random number generator

Variable seed
time&date 12 * + 31 * + 60 * + 60 * + 24 * + seed !
\ not very good entropy gathering ;-)

$10450405 Constant generator
: rol9 ( -- ) dup 9 rshift swap 8 cells 9 - lshift or ;
: rnd ( -- n ) seed @ generator um* drop 1+ dup rol9 seed ! ;
: random ( n -- 0..n-1 ) rnd um* nip ;
```

Die Zahl $\mathcal{N}$ spielt eine größere Rolle: Sie bestimmt den Zahlenraum. Gerechnet wird mod $\mathcal{N}$, alle Zahlen werden also auf den Bereich [0..$\mathcal{N}$-1] zurechtgestutzt. Wir kennen etwas Ähnliches ja von der Integer-Arithmetik, dabei ist $\mathcal{N}$ aber eine Zweierpotenz. Hier ist es ein Produkt zweier Primzahlen.

Die Modulo-Funktion ist eine "teure" Funktion. Effizienter als Divisionen sind Multiplikationen. Wir berechnen deshalb den Kehrwert, und speichern ihn in /N (ohne oberstes Bit). Für halblange Zahlen muss man einen anderen Zähler wählen, und dann auch etwas anders multiplizieren. Diese eine Division spart uns später unzählige wiederholte Divisionen. Man kann die Modulo-Rechnung auch direkt in die Bignum-Multiplikation einbauen. Das ist für Kryptographie-Hardware interessant, weil man hier einen langen Akkumulator benutzen kann, der jede Elementar-Addition gleich mod $\mathcal{N}$ berechnet, und damit auch keine Zwischenergebnisse entstehen läßt, die größer als $\mathcal{N}$ sind. Für Software ist dieser Ansatz aber nicht optimal.

```
Variable N
Variable /N
: d2* ( d1 -- d2 ) 2dup d+ ;

\ reciprocal N computation
: invN ( -- )  -1. N @ ud/mod rot 2drop /N ! ;
: invNh ( -- )
   -1 [ -1 4 cells rshift ] Literal N @ ud/mod rot 2drop /N ! ;
```



Die eigentliche Multiplikation sieht dann so aus. Die Multiplikation mit dem Kehrwert ist nur eine Näherung, wir müssen ggf. etwas korrigieren.

```
\ modulo N multiplication

: um*N ( a b -- c )
   um*
   dup dup /N @ um* rot + nip N @ um* d-
\ corrections
   dup  IF  dup N @ um* d-  THEN
   2dup N @ 0  d< 0= IF  N @ 0 d-  THEN
   2dup N @ 0 du< 0= IF  ." correction failed with N="
         N @ u. cr  THEN
   drop ;
```

Halblang der Einfachheit halber mit `um/mod`. Achtung: Genau dieser Restwert ist für die Performance der Schlüsselerzeugung wichtig. Für reale Anwendungen muss man hier wirklich ausnützen, dass die Zahlen alle nur halb so lang sind wie oben.

```
: um*Nh ( a b -- c )
   um* N @ um/mod drop ;
```

Die eigentliche Verschlüsselungsoperation, die wir anwenden wollen, ist die Potenz. Wir berechnen also a^x mod N . Da alle drei Zahlen sehr groß werden, müssen wir das Problem entsprechend angehen. Die Zerlegung ist vergleichsweise einfach: Für gerade x berechnen wir $(a^{x/2})^2$, für ungerade $(a^{x/2})^2 a$. Das reduziert die Zahl der Multiplikationen bei n bit Schlüssellänge auf maximal $2n$, typisch $1{,}5n$.

```
\ power of n

: **N ( a n -- ) >r 1 swap
  BEGIN  r@ 1 and  IF  dup >r um*N r>  THEN
         r> 1 rshift dup  WHILE  >r dup um*N  REPEAT
         drop drop ;

: **Nh ( a n -- ) >r 1 swap
  BEGIN  r@ 1 and  IF  dup >r um*Nh r>  THEN
         r> 1 rshift dup  WHILE  >r dup um*Nh  REPEAT
         drop drop ;
```

In der Praxis werden diese Primzahlen durch Raten einer Zahl und darauffolgendes Anwenden eines Primzahltests bestimmt.

Wir nehmen hier den verbesserten LUCAS-Test. Der basiert auf dem Fermatschen Satz für Primzahlen:

(a) $a^{N-1} \equiv 1 \bmod N$ und

(b) $a^m \neq 1 \bmod N$ für alle $1 < m < N - 2$

Das sind natürlich etwas viele m. Deshalb gibt es den verbesserten LUCAS-Test, bei dem nur

(b) $a^{(N-1)/q} \neq 1 \bmod N$ für alle Primfaktoren q von $N - 1$

zu berechnen sind (a > 1 ist dabei beliebig).

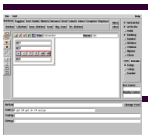
Das Faktorisieren machen wir natürlich nur, wenn der erste Test Erfolg verspricht. Für die Faktorisierung benutzen wir das Sieb des ERATHOSTENES als Hilfsmittel. Endlich kommt dieser altbekannte Forth-Code auch mal zum Einsatz. Für unsere Schlüssellängen (bis zu 64 Bit, also Zerlegungen bis zu 32 Bit) ist das wirklich eine gangbare Lösung. Bei größeren Schlüsseln muss man zu anderen Mitteln greifen, oder Primzahlen verwerfen, derer $N-1$ große Faktoren hat.

```
\ sieve to make factorizing easier

1 2 cells lshift Value maxfactor
$200 maxfactor min Value smallfactor

Create flags maxfactor 2/ 1+ allot
flags maxfactor 2/ + constant eflags

: primes  ( -- n )  flags maxfactor 2/ 1+ 1 fill  0 3 eflags flags
  DO   I c@
     IF  dup I + dup eflags u<
        IF    eflags swap
              DO  0 I c! dup  +LOOP
        ELSE  drop  THEN  swap 1+ swap
        THEN  2 +
     LOOP  drop ;

primes drop
```

Das Sieb braucht man natürlich nur einmal ausrechnen, am besten gleich jetzt zur Compile-Zeit. Kommen wir zur eigentlichen Zahlenzerlegung:

```
\ factorizing

Variable firstfactor

: factorize ( n max -- )  >r  firstfactor @ dp ! 2
   BEGIN  2dup 0 swap um/mod swap 0=  IF  swap rot drop
          dup ,
      ELSE  drop dup 2 =
         IF  1+  ELSE
            BEGIN  2 +  dup 2/ 1- flags + c@  UNTIL
         THEN
      THEN
   2dup u< over r@ u> or UNTIL  drop , here off  rdrop ;

: .factors ( addr -- )
   here swap ?DO  I ?  cell +LOOP cr ;
```

Wir zerlegen die Zahl also von kleinen Faktoren angefangen, bis wir nicht mehr weitermachen brauchen. Eigentlich könnten wir auch den maximalen Faktor mitkorrigieren --- denn der braucht ja nicht größer sein als die Wurzel des derzeitigen Rests.



RSA - eine Modellimplementierung

Kommen wir nun zum eigentlichen Primzahlentest.

```
\ Lucas-test for primes

Variable oN1

: primtest ( a addr -- flag ) >r
   dup  N @ 1- **Nh 1 =  dup 0= IF  rdrop nip  EXIT  THEN
   oN1 @ N @ <> IF  N @ 1- maxfactor factorize  N @
   oN1 !  THEN
   here r> ?DO
      I 2@ <>  IF  over N @ 1- 0 I @ um/mod nip
   **Nh 1 <> and  THEN
      dup 0= ?LEAVE
   cell +LOOP  nip ;
```

Das entspricht der oben angegebenen Formel. Wir können also losgehen und Primzahlen suchen:

```
: primsearch ( -- )
   BEGIN  N @ 2 - random 2 + firstfactor @ primtest
      0= WHILE  2 N +!  invNh  REPEAT ;
```

Damit finden wir recht zuverlässig eine Primzahl. Unsere Primzahlen sollen etwa gleich lang sein, der Modulus sollte im oberen Viertel der Schlüssellänge sein. Das erleichtert das Rechnen.

```
\ find a prime pair

-1 4 cells rshift Constant maxprim

: primsearches ( -- )
   here firstfactor !
   maxprim 3 rshift random
   maxprim dup 3 rshift - +
   -2 and 1+ N !  invNh  primsearch
   firstfactor @ dp ! ;

: primepair ( -- a b )
   primsearches N @  BEGIN  primsearches  dup N @ <>
                     UNTIL
   N @ ;
```

Zur Sicherheit könnten wir noch mit verschiedenen a testen. Das ist zwar bei einer sicheren Faktorisierung wie hier nicht wirklich nötig, aber für längere Schlüssel kann man eben nicht unbedingt zuverlässig faktorisieren (wenn man das effizient könnte, wäre der ganze Witz dieses Verfahrens weg). Wir können also bei unserer Schlüsselgenerierung weitermachen, nachdem wir den komplizierten Teil --- zwei zufällige Primzahlen auszuwählen --- hinter uns gelassen haben.

2. Berechne $\varphi(\mathcal{N}) = (p - 1) \cdot (q - 1)$, wobei $\varphi(\mathcal{N})$ für die EULERsche φ–Funktion steht.

```
\ create a public/private key pair

Variable phiN
Variable d
Variable e

-1 dup 2 rshift xor Constant minN

: genphiN ( -- )
   BEGIN  primepair 2dup *  dup N !  minN u<=  WHILE
   2drop  REPEAT
   invN 1- swap 1- * phiN ! ;
```

3. Wähle eine Zahl $e > 1$, die teilerfremd zu $\varphi(\mathcal{N})$ ist. Dazu brauchen wir den größten gemeinsamen Teiler.

```
: gcd ( a b -- c )  2dup u< IF  swap  THEN
   BEGIN  tuck 0 swap um/mod drop dup 0= UNTIL drop ;

: gene ( -- )
   BEGIN  phiN @ 2/ random  phiN @ 2/ 2/ +
      phiN @ over gcd  1 <>  WHILE  drop  REPEAT  e ! ;
```

4. Berechne die Zahl d so, dass das Produkt $e \cdot d$ kongruent 1 bezüglich des Modulus $\varphi(\mathcal{N})$ ist, dass also $e \cdot d \equiv 1 \bmod \varphi(\mathcal{N})$ gilt.

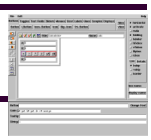
Zur Lösung dieser Diophantischen Gleichung, die nur ganzzahlige Koeffizienten hat, müssen wir eine Kettendivision durchführen (EUKLIDscher Algorithmus, siehe oben gcd). Dieser liefert uns die nötigen Koeffizienten q_n. Die Rücktransformation mit $y_{n+1} = q_n \cdot y_n + y_{n-1}$ mit $y_{-1} = 0$; $y_0 = 1$ liefert uns dann die Lösung (d). Die Rücktransformation erledigen wir einfach nebenher, da ja die einzelnen q_n genau in der Reihenfolge anfallen, in der wir sie brauchen.

```
: gend ( -- )  0 1  phiN @ e @
   BEGIN  tuck 0 swap um/mod  >r
      2swap tuck r> * + 2swap
      dup 0= UNTIL  2drop drop d ! ;
```

Die Zahlen $\mathcal{N}$ und e werden veröffentlicht (Öffentlicher Schlüssel, public key), d, p und q und damit auch $\varphi(\mathcal{N})$ bilden den geheimen Schlüssel (secret key --- streng genommen braucht man nur d).

Jetzt können wir loslegen. Nur noch ein paar Definitionen zur Anwendung:

```
\ encryption/decryption

: encrypt ( a -- b ) e @ **N ;
: decrypt ( b -- a ) d @ **N ;

\ generate key and test if we have a good solution for d and e

$20 Value tests
```



```
: testkey ( n -- flag )
   true  swap 0 ?DO  N @ random dup encrypt decrypt =
      ?code and  dup 0= ?LEAVE  LOOP ;

: genkey ( -- )  genphiN
   BEGIN  gend  gene
   d @ e @ um* phiN @ ud/mod 2drop 1 =  UNTIL ;
```

Oben beim Lösen der Diophantischen Gleichung kommt anscheinend nicht immer ein vernünftiges Ergebnis heraus. Für diesen Fall suchen wir einfach ein neues Paar ***d*** und ***e***.

2 Praktische Erwägungen

Damit dieser Code wirklich verwendet werden kann, benötigen wir einige Dinge:

* Einen richtigen Zufall. Unter Linux gibt es etwa /dev/random. Hier werden verschiedene asynchrone Ereignisse (Benutzereingaben, oder Pakete aus dem Internet) mit den Mikrosekunden der Uhrzeit dieser Ereignisse zu Bits verrührt. Diese Bits sind wirklich nichtdeterministisch, und damit für die Schlüsselerzeugung brauchbar.

* Eine Bignum-Bibliothek. Kern sind Multiplikation, Addition und Division. Die Division kann dabei durchaus etwas langsamer sein. Für das Faktorisieren günstig ist zudem eine gemischte Division, da der Divisor hier nicht groß wird.

* Symmetrische Verschlüsselung für die eigentliche Nachricht.

* Ein sicherer Hash für die Signatur.

Wenn wir den symmetrischen Schlüssel oder den Hash verschlüsseln, sollten wir die restlichen Bits mit zufälligem Müll auffüllen. Das erschwert einem Angreifer einen Rückschluß auf die Daten, die ja (zumindest beim Schlüssel) mehrfach transportiert werden können: Für jeden Empfänger einer. Etwas Bitmüll macht sich auch beim Auffüllen ("padding") der zu verschlüsselnden Nachricht gut. Dadurch sieht auch dieselbe Nachricht mit demselben Schlüssel verschlüsselt immer wieder anders aus.

Betreff: VD als PDF im Internet?
Von: Michael Kalus <michael.kalus@onlinehome.de>

> Hmm - wieviele Leute würden potentiell mit einem 56 kBit
> Modem an einer 10 MByte Datei verzweifeln?

Fragen wir die Mitglieder: Also Leute, wie ist das? Wer braucht die gesammelten VDs aller Zeiten als PDF und komprimiert so klein es überhaupt geht - oder ist es auch ok, wenn die 'wie sie sind' auf der Homepage zur Verfügung gestellt werden? Also um 1-3 MB pro Stück, selten mehr, wenige Hefte bis 10 MB?

Michael Kalus

Antworten bitte an die VD

Wiki Forth

Bernd Paysan

Wiki Forth (WF) ist ein kleines CMS (Content Management System) in Forth. Der Name kommt daher, dass WF eine Spache benutzt, die denen der verschiedenen Wikis ähnlich sieht, also weitgehend normaler Text mit ein paar Steuerzeichen eingestreut. Bis auf ein paar historische Reste ist meine ganze Homepage in WF geschrieben. Die aktuelle Version von WF erzeugt W3C-konformes XHTML mit CSS. Es gibt zwar Browser, die das nicht so richtig können und trotzdem versuchen (“Netzkappe“ 4 und Internet “Exploder“), die Seiten sind aber auch ohne CSS noch lesbar.

Was ist ein CMS?

Ein CMS soll die Verwaltung von Web-Seiten erleichtern. Gerade wenn die Seiten etwas umfangreicher sind, und öfter hier und da etwas geändert wird, ist das sinnvoll. Natürlich sollen Inhalt und Form voneinander getrennt sein. Bei der Eingabe konzentriert man sich auf den Inhalt; die Seiten werden dann entsprechend dem gewählten Stil umgesetzt.

Zudem sollte ein CMS noch Verwaltungsfunktionen anbieten, etwa tote Links erkennen (innerhalb des verwalteten Bereichs), und andere automatisch generierbare Informationen zur Verfügung stellen. Eine sehr wichtige Funktion ist für mich z.B., die Größe einer verlinkten Datei anzugeben.

Die Steueranweisungen

Die Steueranweisungen teilen sich in drei Gruppen auf:

* Globale Informationen über die Seite
* Struktur des Dokuments
* Inline-Elemente (Links, Bilder, Texthervorhebung)

Globale Informationen

Die globalen Informationen sind als Forth-Befehle am Anfang der Seite zu finden. Eine Seite wie diese enthält zumindest einige minimale Informationen über den Maintainer und dessen E-Mail-Adresse, das erste Dokumentdatum und das zugehörige Style-Sheet.

```
maintainer Bernd Paysan
created 11apr2004
css "wf.css"
```

Hat die Seite ein Menü, stellt man es hier zusammen. Menüs sind Links, enthalten also einen Text und eine Datei, getrennt durch das Pipe-Symbol |. Aus dem Text kann man ein Icon generieren lassen.

```
up-toc Home|index.html
top-toc 4stack|4stack.html
top-toc b16|b16.html
top-toc Gforth|gforth.html
top-toc bigFORTH|bigforth.html
this-toc Wiki Forth|wf.html
sub-toc CMS|wf.html#CMS
sub-toc Steueranweisungen|wf.html#ST
```

Die Icons generiert am besten der Gimp. Alle Icons werden im Unterverzeichnis navigate abgespeichert. Dort wird auch eine Gimp-Batch-Datei erzeugt (nav.scm), der nur noch das (gimp-quit 0) als letzte Zeile fehlt. Den Gimp ruft man also auf mit

```
echo '(gimp-quit 0)' >>navigate/nav.scm
gimp -i --batch '(load "navigate/nav.scm")'
```

Als Basis für die Generierung des Navigationsbuttons dient die Datei navigation.scm - die muss man in das scripts-Verzeichnis vom Gimp kopieren. Als leeren Button verwendet die eine Datei button.jpg, deren Pfad in navigation.scm drin steht. Als Font wird der Blippo-Font aus dem Freefont-Packet verwendet. Man kann natürlich auch einen der vielen bereits vorhandenen Button-Funktionen im Gimp nehmen, aber Gimp-Programmierung ist jetzt ein anderes Thema.

Die eigentliche Seite wird mit dem Wort wf eingeleitet. Dem folgt die erzeugte HTML-Datei und der Seitentitel. wf scannt die Eingangsdatei, bis es auf einen einzeln stehenden Punkt trifft; hier endet die Seite.

```
wf wf.html "Wiki Forth"
```

Struktur des Dokuments

Die Steueranweisungen für die Grobstruktur orientieren sich am Emacs-Outline-Mode. Überschriften beginnen mit ein, zwei oder drei Sternen (h1 bis h3 in HTML). Jedem Element kann man ein Label zuordnen, indem man && label in die Zeile davor stellt. Horizonale Leisten erzeugt man mit ---. Untermenüs mit -- label, hier werden die Sub-Menüs wieder erzeugt, und das aktuelle markiert.

Natürlich kann man auch Listen anlegen. << startet eine Liste, >> beendet sie. Ob die Liste mit Punkten oder Nummern markiert wird, entscheidet das Startzeichen: - für Punkte, + für Nummern. Die Zeichen ? und : sind für descriptive Listen reserviert, wobei ? einen Titel einleitet, und : die eingerückte Erklärung.

Wenn die Struktur über mehr als einen Absatz gehen soll, hängt man ans Startzeichen noch zwei << dran, und beendet das Ganze mit >>. Dabei sind natürlich wieder Unterstrukturen erlaubt.

Unformatierten ASCII-Text (für Listings) bindet man mit :code .. :endcode ein, oder mit :code-file datei gleich eine ganze Datei.

Implementiert werden diese Wörter alle in einer Wordlist namens longtags. Interpretiert werden sie, wenn sie am Anfang eines Absatzes stehen. Steht kein verwertbares Wort am Anfang eines Absatzes, wird er als ganz normaler Absatz gesetzt:

```
: section-par ( -- )  >in off
   bl sword longtags search-wordlist
   IF    execute
   ELSE  source nip IF  >in off s" p" par  THEN  THEN ;

: parse-section ( -- )  end-sec off
   BEGIN  refill  WHILE
  section-par end-sec @ UNTIL  THEN ;
```

Inline-Elemente

Innerhalb des Texts will man natürlich auch noch Variationen verwenden. Etwa **fette** oder *hervorgehobene* Texte. Oder `Schreibmaschinenschrift`. Natürlich auch noch Links oder Bilder oder Bilder im Link.

Diese Tags bestehen aus einzelnen Buchstaben am Anfang eines Wortes. * steht für fett, _ für unterstrichen, und # für Schreibmaschinenschrift. Wer einen dieser reservierten Buchstaben braucht, kann sie in eine Tilde ~ einfassen.

Bilder haben folgende Syntax: {Text|URL}. Links sind fast gleich aufgebaut: [Text|URL], allerdings darf der Text auch Bilder enthalten. Ein paar Sonderzeichen direkt nach dem | werden in Optionen umgesetzt.

Link-Optionen

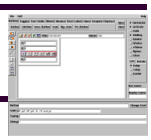
Links werden normalerweise automatisch mit einem Icon versehen. Icons kann man für ganze URL vergeben, oder für bestimmte Suffixe. Bei der Suche nach dem Icon wird die URL stückweise nach . durchsucht, und der Rest als Dateiname im Unterverzeichnis icons mit angehängtem .* gesucht. Das gefundene Icon (GIF, PNG oder JPEG) wird dann vor den eigentlichen Link-Text gesetzt. Wer das verhindern will, setzt als Options-Zeichen ein "\".

Längere Dateien (für Downloads) sollte man immer mit der Größe versehen. Dafür ist die Link-Option % vorgesehen. Die hängt die Größe der Datei in Kilobytes (oder Megabytes, wenn's mehr als 2 sind) hinter den Link (in Klammern). Natürlich kann man beide Link-Optionen gleichzeitig verwenden, die Reihenfolge spielt dann keine Rolle.

Ich signiere Software-Downloads. Auch hier gibt es einen Automatismus: Wenn es zum Link eine zugehörige .sig-Datei gibt, wird der Link auf diese Datei automatisch eingebunden.

Bilder-Optionen

Hier gibt es mehr Optionen, und die Reihenfolge spielt auch eine Rolle. Zunächst einmal kann man Bilder ausrichten.



l, <
: Bild wird links ausgerichtet (class "left", genaues bestimmt das Style-Sheet).

r, >
: Bild wird rechts ausgerichtet (class "right")

c, =
: Bild wird zentriert (class "center")

~
: Bild wird zentriert (class "middle")

Normalerweise bekommen Bilder in Links immer einen Rahmen. Will man das verhindern (oder einen schmaleren Namen verwenden), muss man das natürlich auch deklarieren.

-
: Bild bekommt gar keinen Rahmen (class border0)

+
: Bild bekommt einen Rahmen der Größe 1 (class border1)

Allen Bildern wird automatisch die Größe mitgegeben, damit der Browser die Seiten schon komplett rendern kann, bevor die Bilder geladen sind.

Tabellen

Tabellen sind ein Thema für sich, schließlich erlaubt HTML sehr viel mit Tabellen anzustellen. Zur Zeit sind nur die wichtigsten Features implementiert. Die Formatierung lehnt sich dabei an LaTeX an.

Zunächst fangen Tabellen mit **<| Format [Border]** an. Die Format-Zeichen gelten für die jeweilige Spalte der Tabelle, und bedeuten folgendes:

l, < left alignment
r, > right alignment
c, = center alignment

Zeilen innerhalb der Tabelle sind eingerahmt, je nach Bedeutung:

`+| .. |+` Zeile enthält table headers (Kopfzeilen)
`-| .. |-` Zeile enthält normale Tabelleneinträge
`=| .. |=` Zeile beginnt mit einer Kopfzeile, normale Tabelleneinträge folgen

Einträge, die über mehrere Spalten gehen, benötigen eine eigene Formatanweisung. Diese beginnt mit / für Zellen, die mehrere Zeilen überspannen, \ für mehrere Spalten. Die folgende optionale Ziffer gibt die Zahl der Zeilen oder Spalten an (Default ist eine Zeile/Spalte). Danach kann noch ein Formatzeichen kommen (siehe oben). Es ist möglich, beide Formatanweisungen zu verwenden, wobei zuerst / und dann \ kommen muss.

Beispiel:

Das kann man natürlich nur an einem Beispiel zeigen. Eine einfache Tabelle sieht etwa so aus:

```
<| clrr 1
+| Menge | Artikel | Preis | Summe |+
-| 3      | Äpfel   | 0,30  |  0,90 |-
-| 1      | Banane  | 0,50  |  0,50 |-
=| \3l Gesamtpreis          |  1,40 |=
|>
```

Menge	Artikel	Preis	Summe
3	Äpfel	0,30	0,90
1	Banane	0,50	0,50
Gesamtpreis			1,40

Eine komplexe Tabelle zeigt alle Möglichkeiten:

```
<| rccl 2
+| /3 1 | 2 |\2c 40  |+
-| /3\2c center | 6  |-
-|                8  |-
-| /2 9 |         1  |-
=| \2r 12   |/r   4  |=
-| 1    | 2 | 3 | 40 |-
-| 1234 | 1243 | 1234 | 1234 |-
|>
```

<table>
<tr><td rowspan="3">1</td><td colspan="2">2</td><td>40</td></tr>
<tr><td rowspan="2" colspan="2">center</td><td>6</td></tr>
<tr><td>8</td></tr>
<tr><td rowspan="2">9</td><td></td><td></td><td>1</td></tr>
<tr><td colspan="2">12</td><td>4</td></tr>
<tr><td>1</td><td>2</td><td>3</td><td>40</td></tr>
<tr><td>1234</td><td>1243</td><td>1234</td><td>1234</td></tr>
</table>

Man muss bei solchen komplexen Tabellen natürlich immer die ausgelassenen Tabellen im Kopf behalten.

Komfortfunktionen

Neben der eigentlichen Texteingabe braucht ein CMS natürlich noch mehr Komfort.

Autoreplaces

So sollen vielleicht bestimmte Schlüsselwörter immer als Link erscheinen (wie bei einem Wiki). Dazu gibt es autoreplaces. Diese Wörter werden zu Links expandiert, wann immer sie im Text auftauchen. Beispiel:

autoreplace ForthGesellschaft [Forth Gesellschaft|http://www.forth-ev.de/]

ersetzt jedes Vorkommen von "ForthGesellschaft" durch den entsprechenden Link Forth Gesellschaft .

Datenbank

Man kann sich natürlich noch komplexere Einträge vorstellen, als einfache Links. Das braucht sowas wie eine einfache Datenbank. Zugriff hat man auf den Eintrag über das Schlüsselwort, eingefügt wird ein Text, der sich aus den einzelnen Feldern zusammensetzt. Ich verwende diese Datenbank an zwei Stellen: Für meine Changelog-Einträge in bigFORTH, und für Projektmitarbeiter auf den Intranetseiten in der Arbeit (die Mitarbeiter haben immer dieselben Daten, die man natürlich nicht für jedes Projekt nochmal eingeben will). Hier will ich nur die Changelog-Datenbank erklären. So ein Eintrag sieht etwa so aus:

```
change: 01sep2003
Version: 2.0.11
minos: 1.0.0
text: fixes a few bugs, like FORTH-WORDLIST being
commented out, and problems with read only filesy-
stems (like Knoppix). Also fixed OpenGL bindings,
created a new setup script for Inno Setup 3.x, and a
windows distribution. Fixed Makefile and configure
process so that the windows version can be compiled
from source with #./configure; make# when cygwin is
installed. Bumped up MINOS version to 1.0.0 (there
are little further changes to expect).
.
```

Das Datum dient dabei als Schlüsselwort. Der Code dazu ist auch nicht so schwer. Zunächst einmal definiert man sich die Tabelle (aus Text-Feldern und Absätzen), die Ausgabe-Prozedur ist ein deferred Word, weil sie natürlich erst später definiert werden kann:

```
Defer .log
' .log 4 table: change:
field: version:
field: minos:
par: text:
```

Das war noch einfach. Die Tabelle besteht aus vier Feldern mit den Namen change:, version:, minos: und text:. Beim Aufruf wird .log ausgeführt. Was soll da passieren? Es sollen die Versionen mit dem Wort ?rev ausgegeben werden, und der Text als solcher (ohne besondere Methode; die Datenbank weiß, wie man Paragraphen ausgibt). db-par scannt durch den Text bis es an eine Punkt am Zeilenanfang ankommt. Variablen stehen in %, werden sie nicht mit der Standardmethode ausgegeben, muss der gewünschte Forth-Code mit | abgetrennt werden (es ist auch mehr als ein Befehl erlaubt).

```
: .rev ( addr u -- ) s" b" tagged ;

: ?rev ( addr u -- ) dup 0= IF  2drop  EXIT  THEN  .rev ;

: .entry-rest ( addr -- )
   db-par The version %version:|?rev% from %change:|?
   rev% %text:%
. ;

: .entry-par ( addr -- ) LT -<< .entry-rest LT >> ;

' .entry-par is .log
```

Als Besonderheit gibt .last noch den zuletzt definierten Changelog-Eintrag etwas anders formatiert aus:

```
: .last ( -- ) last-entry @
   db-par Current version is bigFORTH %version:|.rev%,
   Minos beta
   %minos:|.rev%, from %change:|.rev%.
. last-entry @ .entry-rest ;
```

Bernd Paysan
01/04.2004

Gehaltvolles

zusammengestellt und übertragen
von Fred Behringer

VIJGEBLAADJE der HCC Forth-gebruikersgroep, Niederlande

Nr. 46, Oktober 2004

Forth vanaf de grond 4
Ron Minke

Vierter Teil des Tutorials "Forth von Grund auf". Am Ende des Artikels steht "wird fortgesetzt". Ich (der Rezensent) gebe die Vorrede des Autors wieder:

"Im vorausgegangenen Beitrag hatten wir versucht, je ein AVR-Registerpaar den Zeigern IP und SP zuzuweisen. Beim Auskundschaften der sich dafür anbietenden Möglichkeiten haben wir die Verwendung des AVR-Registerpaars Z festgelegt. Wir verwenden es als Notizblock, als Zwischenspeicher, der schließlich zum Ziel hat, den indirekten Sprung auszuführen."

Und weiter in den Festsetzungen 6 und 7: "Der Datenstack-Zeiger SP wird dem AVR-Registerpaar Y zugeordnet, der Interpreter-Zeiger IP dem AVR-Registerpaar X. Wir können nun den fast schon endgültigen Code für NEXT zusammenstellen."

ciasdis, een reverse engineering tool in Forth
Albert van der Horst

Aus der Vorrede: "Es ist ein Krieg im Gange. Es geht darum, ob das Wissen, das die Menschheit mit noch nie dagewesener Schnelligkeit ansammelt, in den Händen einiger weniger bleibt oder jedem zur Verfügung gestellt werden soll. Eine bedeutsame Rolle in diesem Krieg spielen die Reverse-Engineering-Tools. Mein ciasdis ist so ein Tool. Der englische Ausdruck "Reverse Engineering" bedeutet soviel wie "den Entwurf zurückverfolgen". Es geht darum, aus etwas Bestehendem (ein Auto, eine Schaltung, ein Programm) die Kenntnis über dessen

Zusammensetzung zu rekonstruieren. Das Programm mit Quelltext kann von **http://home.hccnet.nl/a.w.m.van.der.horst/forthassembler.html** heruntergeladen werden."

Der Rezensent: Diesen Artikel halte ich für sehr interessant. Er stimmt mich nachdenklich. Wenn wir als Kinder Uhren auseinandergenommen haben oder nach Kriegsende aus zurückgelassenen Wehrmachtsbeständen elektronisch aussehende Dinge, dann war das also eine Vorstufe des "Reverse Engineering"? Uns ging es natürlich nur um "Wissensdrang" (profan gesagt, um jugendliche Neugier), nicht um Rekonstruktion. - Den ersten japanischen Fotoapparaten, die auf den deutschen Markt kamen, hat man (in maßloser Selbstüberschätzung) nachgesagt, dass sie auf Reverse-Engineerings-Wegen entstanden seien. Heute scheint es wohl umgekehrt zu sein? - Vieles, was ich über Forth weiß, habe ich auf dem Wege des Reverse-Engineerings gelernt. - In der Statistik versucht man, aus einem Wust von Daten allgemeine Zusammenhänge herauszulesen. "Reverse Engineering"? - In der Theorie der Reellen Funktionen gibt es den Begriff des "Level-Sets" (zur Charakterisierung quasikonvexer Funktionen). Ist dessen Ermittlung "Reverse Engineering"? Ganz allgemein die Bestimmung des Urbildes (nichtinjektiver) Funktionen? In der Optimierungstheorie interessiert man sich für die "Menge der optimalen Punkte", also derjenigen Punkte aus der "zulässigen Menge", die über die "Zielfunktion" zum "Zielwert" führen. Reverse Engineering? - Ulrich Paul hat in der VD über assoziative Speicher (Auffinden aller Adressen, die mit einem vorgegebenen Wert belegt sind) geschrieben. Könnten wir nicht mal auch einen Beitrag zum Reverse Engineering lesen? - Provokativ interpretiert: "Engineering" = Herstellung eines schwarzen Kastens, der bei bestimmter Belegung der Eingänge bestimmte Belegungen der Ausgänge erzeugt. "Reverse Engineering" = Nachmachen eines schwarzen Kastens unbekannter Herkunft und unbekannter Konstruktion, der bei gleicher Eingangsbelegung dieselbe Ausgangsbelegung wie das Original erzeugt. - Coca Cola herzustellen, dürfte verhältnismäßig leicht sein, wenn man die Formel kennt. Coca Cola nachzumachen, ist, wie man nachlesen kann, unendlich viel schwerer. -- Ende der Gedankenflüge des Rezensenten, die durch diesen wirklich interessanten Artikel des (als Entwickler von ciForth bekannt gewordenen) Autors ausgelöst wurden.

Adresse der Redaktion des Vijgeblaadjes:
a.nijhof@kader.hobby.nl

Nr. 47, Dezember 2004

Forth vanaf de grond 5
Ron Minke

Fünfter Teil des Tutorials "Forth von Grund auf". Am Ende des Artikels steht "wird fortgesetzt". Ich (der Rezensent) gebe die Vorrede des Autors wieder:

"Im vorausgegangenen Beitrag haben wir IP und SP den AVR-Registerpaaren X und Y zugewiesen. Im vorliegenden Teil übersetzen wir den zur Behandlung von High-Level-Worten verwendeten Pseudo-Code in AVR-Maschinensprachbefehle."

Es ist von W die Rede, und von DOCOL, DOCON, DOVARIABLE etc. Der Atmel-Prozessor AVR unterliegt der Harvard-Architektur (Speicher für Befehle und Daten getrennt). Hier bietet sich für Forth nur das klassische Modell an, die indirekte Fädelung.

ATS: hoe zet ik m'n programma in ROM?
Willem Ouwerkerk

Ich (der Rezensent) gebe die Vorrede ("Abstract") des Autors wieder:

"Das Forth-System auf der ATS-Platine kennt zwei Methoden, generierte Binärdateien in den Computer zu transportieren, um sie später wieder auf der Platine zu verwenden.

Die einfachere ist der Intel-Hex-Download. Dabei wird der generierte Code als Intel-Hex-Datei aufbewahrt. Diese Datei kann ganz einfach wieder zurückgeschickt werden und nimmt dann wieder ihren Platz im RAM ein.

Die zweite Methode macht vom eingebauten Relocator Gebrauch. Dieser passt die Intel-Hex-Datei so an, dass sie in das für eigene Programme vorgesehene 8-KB-Fenster geschoben und in EPROM gebrannt werden kann. Der Binärcode wird dann ein fester Bestandteil Ihres Forth-Systems auf der ATS-Platine."

Der Artikel befasst sich hauptsächlich mit dem relozierbaren Compiler.

FORTHWRITE der FIG UK, Großbritannien

Nr. 127, Dezember 2004

2 Editorial
Graeme Dunbar <g.r.a.dunbar@rgu.ac.uk>

Graeme bedauert, dass es mit dem vorliegenden Heft so lange gedauert hat. Es ist das letzte Heft, das er herausgibt. Chris Jakeman wird das Amt des Redakteurs nicht wieder aufnehmen. Sein neuer Job als Dozent und seine Ambitionen, in die Forschung zu gehen, lassen das nicht zu. Graeme war ursprünglich nur "eingesprungen". Ab jetzt wird immer nur für jeweils drei Hefte versucht, einen Redakteur zu finden. Die nächsten drei Hefte wird Jeremy Fowell, der Chairman, herausgeben.

3 Forth Interest Group UK AGM 2004
Neville A. Joseph

Einladung zur Jahresversammlung (AGM) am 30. Oktober 2004 (aufgrund der redaktionellen Schwierigkeiten mit extremer Verspätung hiermit veröffentlicht) und Kassenbericht. Die neue Führungsriege ist, abgesehen von den Veränderungen in der Redaktion ("rotierende" Redakteure), die alte. Der Kassenbericht erstreckt sich über das Rechnungsjahr vom 1.4.2003 bis zum 31.3.2004. Der Rezensent: Der "jährliche" Mitgliedsbeitrag basiert auf 6 Forthwrite-Heften pro Jahr. Auf jedem Heft steht, wie viele Hefte noch geliefert werden (xx left), bevor der Mitgliedsbeitrag erneuert werden muss.

5 AGM Report
Douglas Neale

Nur ganz wenige Teilnehmer. Die zukünftigen AGMs (Annual General Meeting) werden elektronisch abgehalten. Die Forth-CD ist noch nicht fertig. Arbeiten daran werden aber jetzt abgeschlossen. Kostet 5 englische Pfund. John Milner hat vor, den einen oder anderen Artikel aus der VD für die Forthwrite zu übersetzen. Jeremy Fowell war von einem VD-Heft, das er sah, derart beeindruckt, dass er Mitglied der Forth-Gesellschaft wurde. Für das 25. Jubiläum wurde und wird nichts unternommen. Für den Editor gilt ab jetzt das Rotationsprinzip: Alle drei Hefte ein anderer. Mitgliederschwund um 20%: Nur noch 80 zahlende Mitglieder.

6 A Brief Introduction to FSharp
Jim Lawless

Es geht um F#, auch als weforth bekannt. Jim war auf der Suche nach einem einfachen Forth, das auf das Windows-API zugreifen kann. Er will mit TCP/IP experimentieren. Siehe auch http://www.eforth.com.tw/academy/zip/weforth.zip . Eine ältere Version: http://www.forth.org/svfig/Win32Forth/weforth203.zip

Der Rezensent: Sollen wir im Deutschen "Fis" statt "FSharp" sagen? Natürlich nicht! Kann eine im Sinne der KI durchgeführte automatisierte Übersetzung (jemals) eine derartige Entscheidung fällen?

8 Paths and Brushes
David R. Pochin

David setzt seine einführenden Betrachtungen über die Programmierung von Graphik in Win32Forth fort: In der Datei dc.f gibt es eine Reihe von Worten, die sich auf "Paths" beziehen. Eine ganze Reihe von eigentlich benötigten Worten aber, die in Windows 98 (11) und NT (20) vorgezeichnet sind, fehlen einfach. Ein "Path" wird über eine Folge von Worten zwischen BeginPath und EndPath erzeugt. Diese Folge wird aber weitestgehend dem Programmierer überlassen. MoveTo, LineTo und Ähnliches fehlen. David macht Programmvorschläge.

David, als Forthwrite-Autor wohlbekannt, er war auch in unserer internationalen WebForth-Entwicklungsgruppe, ist inzwischen im Ruhestand. In "seinem früheren Leben" fuhr er zur See, bevor er Dozent für elektronische Navigationsmittel wurde.

15 What Languages Fix
Graeme Dunbar

Diesmal ein Beitrag von James Boyd zur Charakterisierung von Forth: Forth allows prototyping (Forth lässt eine schnelle Entwicklung zu).

16 FIG UK Blog
Jenny Brien

Jenny Brien hat ein Blog unter dem Namen "Forthwith" eingerichtet. Näheres siehe http://figuk.blogspot.com/ .

17 EKEY And Events
Jenny Brien

Jenny bezieht sich auf das, was im ANS-Standard http://www.taygeta.com/fgorth/dpansa10.htm über EKEY und Events gesagt wird, und macht Verbesserungsvorschläge.

20 Across the Big Teich
Henry Vinerts <Volvovid@aol.com>

Henrys Berichte von August bis November 2004 über SVFIG-Aktivitäten in Originalfassung. Wir kennen sie in der Übersetzung von Thomas Beierlein. Henry bestätigt den Empfang des Päckchens mit unseren SWAP-Drachen-Anstecknadeln. Er hat sie "gleichmäßig verteilt".

24 Deutsche Forth-Gesellschaft

Hier wieder unsere Anzeige zur Anwerbung von Mitgliedern für die FG. Alle Angaben sind schon seit einiger Zeit aktualisiert: Preise in Euro und neue Adresse des Forthbüros. Über den Sinn dieser Anzeigen, deren Gegenstücke wir regelmäßig in der VD bringen, kann man sich streiten. Ganz unsinnig sind sie nicht: Jeremy Fowell, der Chairman von FIG UK, ist nicht zuletzt auf diesem Wege Mitglied der FG geworden.

25 Vierte Dimension 2/2004
Joe Anderson <jia@jus.abel.co.uk>

Joe bespricht das Heft 2/2004 unserer Vierten Dimension. Die englischen Forth-Freunde hatten Schwierigkeiten, das Redak-

teursproblem zu lösen. Umso höher ist es Joe anzurechnen, dass er an den Heftbesprechungen (Übersetzungen) beständig weiterarbeitet.

27 Wanted
Chris Jakeman <cjakeman@bigfoot.com>

Chris sucht für seine Lehraufgaben am Peterborough Regional College in Computergeschichte: 5,25"- und 8"-Disketten, Lochstreifen, Lochkarten, Kernspeicher und transistorbestückte Prozessor-Karten aus Restbeständen, vom Speicher und vom Keller.

28 FIG UK Contacts and Information and Services to Members

Die Mitgliedschaft kostet 12 englische Pfund pro Jahr. Sie berechtigt zu sechs Forthwrite-Heften. Chairman und Editor (für die nächsten drei Hefte): Jeremy Fowell, Sekretariat: Douglas Neale, Webmaster: Jenny Brien, Literaturverleih: Graeme Dunbar.

Auf die Frage von Rolf Schöne, ob Rolf Lauer das VFX von Stephen Pelc nutze, und wie Rolf Lauer auf die FG aufmerksam geworden ist, gab es folgende Antwort:

Guten Morgen Herr Schöne,

ja genau, Stephen Pelc. Leider ist sein Handbuch nicht proportional zur Güte des Systems. Anyway, man kann nicht alles haben :-) Ich gehe mal davon aus, daß die meisten Mitglieder wohl ausschließlich Freeware einsetzen, und es keinen weiteren Benutzer von VFX Forth gibt. Ist das richtig ?

Wie bin ich auf Forth-e.v. aufmerksam geworden ?
Ehrlich gesagt, ich weiß es nicht mehr. Der Forth-e.v. ist mir schon sehr lange ein Begriff und vor 3 oder 4 Jahren wollte ich schon mal Mitglied werden. Aber dann kam mal wieder etwas dazwischen und die Arbeit fraß mich auf und ich vergaß es wieder. Letzte Woche dachte ich dann: Wenn alle so träge sind wie ich, ist der Verein irgendwann dicht. Dann füllte ich das Formular aus und schickte es weg.

Auf jeden Fall freue ich mich diesen Schritt endlich getan zu haben!

Mit den allerbesten Grüßen aus dem Odenwald

Rolf Lauer

Wir freuen uns ebenfalls, Rolf Lauer. Und ich hoffe, daß, so vorhanden, andere VFX-Nutzer bald über das Forthbüro oder die VD Kontakt mit Ihnen aufnehmen.
Mit einem freundlichen Glückauf

Friederich Prinz

Betreff: **Thinking Forth Nachdruck**
Von: **Bernd Paysan** <bernd.paysan@gmx.de>

Und noch eine Ankündigung: Die Arbeiten am Nachdruck von "Thinking Forth" sind fertig. Man kann das Buch jetzt offiziell herunterladen

http://thinking-forth.sourceforge.net/

oder bei Amazon.com eine Tote-Bäume-Version bestellen (Print on Demand).

http://www.amazon.com/exec/obidos/tg/detail/-/0976458705/qid=1105747164/sr=8-1/ref=sr_8_xs_ap_i1_xgl14/103-1076596-4278268?v=glance&s=books&n=507846

Das Erscheinungsbild dieser Amazon-Seite dürfte sich in den nächsten Tagen noch ändern, wenn Leo herausgefunden hat, wie man das Cover-Bild anzeigt, und den Kunden im Text suchen lässt.
Bis zu Amazon.de hat das Buch noch nicht hingefunden. Dort sind noch ab-85-Euro-Angebote der Erstauflage zu haben (mit allen Dreckfuhlern, die wir inzwischen beseitigt haben - aber ohne die Dreckfuhler, die wir hinzugefügt und übersehen haben ;-) .
Nachdem der Nachdruck fertig ist, können wir uns auch an die nächste Phase dieses Projekts wagen: dem Aktualisieren und Ergänzen (und ggf. auch Übersetzen). Das, was ich auf meiner Agenda habe, steht auch auf der TF-Seite (priorisiert):

* Die Beispiele so umschreiben, dass sie mit aktuellen ANS Forth-Systemen laufen
* Den Coding Style auf aktuelle Gepflogenheiten (lower case) anpassen
* Kapitel über Forth und OOP, Forth debugging und Programmpflege hinzufügen
* Forth-Denker interviewen, die vor 20 Jahren nicht dabei waren
* Übersetzen (nicht in C, sondern z.B. auf Deutsch)

Und natürlich stärkt das alles unsere Position bezüglich eines Starting-Forth-Updates.

Bernd Paysan

Von: **Carsten Strotmann** <cas_news@strotmann.de>
...
tolle Arbeit.

In DE lässt sich diese Buch bei Bookzilla/Libri schon (vor-) bestellen.
Nicht auf Amazon warten. Und dann gehen auch die Prozente an die FSF (Free Software Foundation), anstatt an Amazon.

http://www.bookzilla.de/shop/action/productDetails?aUrl=90006951&artiId=3149667

Ciao, *Carsten*

ARINC 429

Rafael Deliano

Obwohl die meisten Leute nur CAN kennen, hatten manche "embedded" Systeme schon viel früher serielle Bussysteme. Hier gebe ich einen Überblick über den Standard in zivilen Flugzeugen; seit 1977 ab Boeing 737 und Airbus A320, 340 und einigen Hubschraubern in Verwendung.

Die Leistungsfähigkeit reicht aber heute nicht mehr so richtig. Boeing hat deshalb in der 777 ein proprietäres, patentiertes Bussystem eingeführt [3]. Wirtschaftlich keine gute Idee, so daß das System nachträglich geöffnet wurde und heute als ARINC 629 bzw. DATAC bekannt ist. Allerdings ohne, daß sich ein offener Zuliefermarkt entwickelt hat. Airbus steht vor dem gleichen Problem. Hier scheint die Lösung ab A380 die Ethernet-Variante AFDX zu sein.

Es ist also kein direkter Nachfolger in Sicht und die Lebensdauer von Flugzeugen ist hoch. Viele Verkehrsflugzeuge sind älter als 20 Jahre. Beides hält den Oldtimer ARINC 429 am Leben. Es existieren in Großflugzeugen ohnehin oft mehrere Bussysteme für unterschiedliche Aufgaben parallel. Die 777 hat außer ARINC 629 deshalb auch noch ARINC 429. Der Militärtransporter C-17 neben ARINC 429 den MIL-STD-1553. Und auch bei Airbus werden wohl solche Mischlösungen weiterexistieren.

ARINC ist die „Aeronautical Radio INCorporated" [1], die das Copyright auf die Spec hat und sie kostenpflichtig verbreitet. Eine besser lesbare Kurzfassung ist [2].

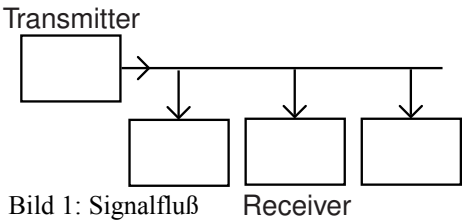


Bild 1: Signalfluß

Die Bezeichnung "Datenbus" scheint ohnehin etwas übertrieben, da nur in eine Richtung Daten an bis zu 20 Empfänger übertragen werden (Bild 1).

Trotzdem war selbst dieser simple Multiplexbus für den Anwender ein enormer Fortschritt.

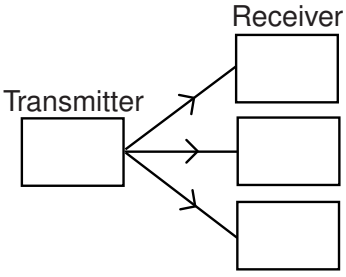


Bild 2: altern. Verdrahtung

Alte L-1011 Verkehrsflugzeuge aus den 70er Jahren hatten 240 km Kabel installiert [9]. Die Mondlandefähre von Apollo 11 von 1969 war mit 47 km vollgestopft. Auch Schiffe wie ein 350 tdw Tanker von 1975 waren mit 30 km verkabelt.

Für bidirektionalen Datenaustausch müssen zwei Busse installiert und verkabelt werden. Geräte haben oft 2 bis 8 Sender und 2 bis 3 mal soviel Empfänger. Die typische Bezeichnung für Schachteln ist LRU „Line Replacable Unit". Elektrisch günstiger als die Verdrahtung in Bild 1 scheint zwar Bild 2. Wird wegen des erheblichen Aufwands an Kabeln heute aber eher gemieden.

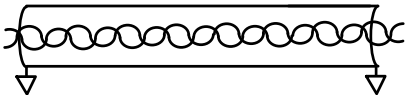


Bild 3: Kabel

Kabel

Vorgesehen ist ein verdrilltes, geschirmtes Kabel mit 75 Ohm Impedanz. Der Schirm wird an allen Endpunkten und Abzweigungen geerdet. Man beachte, daß für 400 Hz Umgebung mehr twists für die Verdrillung wünschenswert sind als bei 50 Hz.

Die Ausgangsimpedanz des Treibers beträgt 75 Ohm, um das Kabel korrekt zu terminieren. Die Empfänger haben mehr als 8k Impedanz. Bei den maximalen 20 Empfängern also 400 Ohm Last.

Die Kabellänge ist nicht direkt spezifiziert. Literaturangaben geben 300 bis 500 Fuß an, was in einer Boeing 747 wohl vorkommen kann. Typische Systeme haben eher weniger als 175 Fuß. Gut für Reichweite sind wenig Empfänger. Gutes Kabel. Gute Baugruppen, die möglichst wenig Eingangskapazität haben (vgl. Dioden für Blitzschutz). Prüfung der Signalform mit Oszilloskop am installierten System ist wohl immer nötig. Der weite Temperaturbereich verändert die Charakteristik der Endgeräte etwas. Kabel in Flugzeugen altert [11]: Isolierung wird brüchig, kann feucht werden und damit Impedanz verschlechtern. Die Signalform sollte also gut in der Toleranz liegen, um akzeptabel zu sein.

Treiber

Die beiden Kabel werden gegenphasig mit einem ternären Signal angesteuert (Bild 4). Ternär spart mit dem Ruhepegel NULL Strom. Es gibt zwei genormte Geschwindigkeiten: 12,5kHz (12 - 14,5kHz) und 100kHz +/- 1%. Dementsprechend

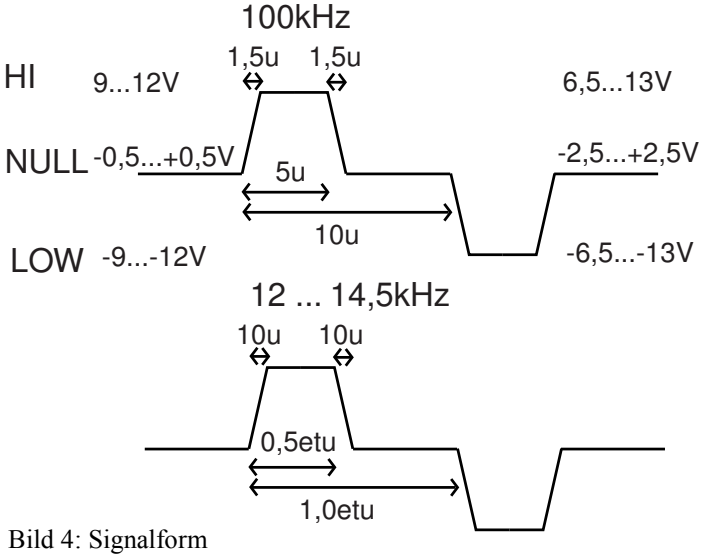


Bild 4: Signalform

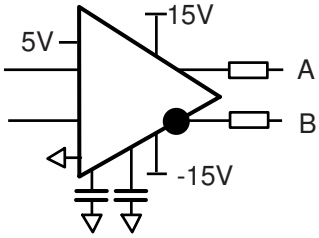


Bild 5: Pegelumsetzer Sender

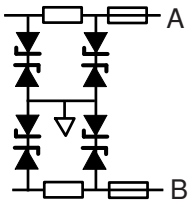


Bild 6: Serienwiderstände für Blitzschutz ausgerüstet

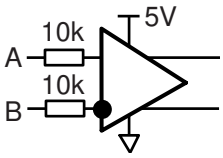


Bild 7: Pegelumsetzer Empfänger

gibt es Unterschiede fürs Timing. Die Begrenzung der slewrate vermindert Reflexionen auf dem Kabel und Abstrahlung.

Ein breites Angebot an ICs für ARINC 429 hat [4] und dort ist auch eine Crossreferenz verfügbar, die das Angebot von [4] bis [8] darstellt. Der Leitungstreiber (Bild 5) war ehedem der Standardtyp „3182“. Das IC hat extern zwei Kerkos, mit denen die fallende und steigende slewrate programmiert wird. Bei einigen Hybridschaltungen sind diese anscheinend mit Fets schaltbar, so daß man per Software zwischen beiden Geschwindigkeiten wählen kann. Es gibt sogar Treiber, die man auf RS-422 umschalten kann [5], um Produkte noch flexibler zu machen.

An den Kabeln in Flugzeugen ist Schutz gegen Blitzschlag gefordert. Wegen der niedrigen Ausgangsimpedanz ist das am Sender schwierig. Wie in Bild 6 angedeutet, sind deshalb externe Serienwiderstände, die auf 75 Ohm ergänzen, externe Schutzdioden und Sicherungen üblich.

Das nächste Problem ist die Verlustleistung, besonders bei Kurzschluß gegen Masse oder zwischen den Leitungen. Bei -55 ... +125’C Umgebungstemperatur ist das auch mühsam. Deshalb ehedem Gehäuse mit vielen Pins wie 16 Pin CERDIP, heute Kühlflächen bei SMD-Bauformen. Absenken der Versorgungsspannung auf +/-12V hilft auch, aber beschränkt die Zahl der Empfänger und die Reichweite.

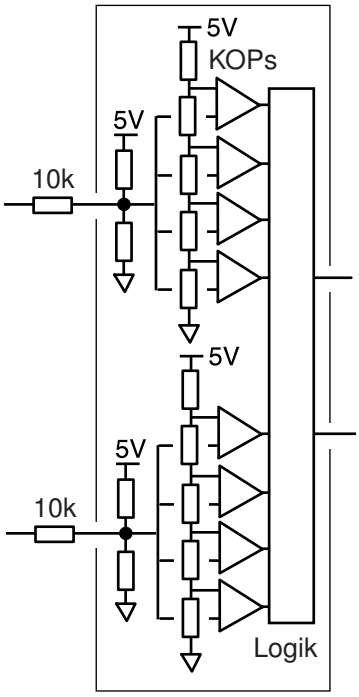
Der Empfänger (Bild 7) hat keine Probleme mit Verlustleistung und gegen Blitz genügt oft schon ein externer Serienwiderstand. Durch einen passend ausgelegten internen Spannungsteiler reichen 5V Versorgungsspannung (Bild 8), die KOPs untersuchen dann, ob die Pegel passen. Die tatsächliche Innenbeschaltung aktueller ICs ist komplizierter, um auch Gleichtaktfehler ausgleichen zu können.

Die Funktion von Interface-ICs wie den 3183 bzw. 3283 wurde bald in die „UART“s integriert.

„UART“

Hinter den Treibern kommen CMOS-LSIs, die die unterste Ebene des Protokolls machen. Heute können das auch ASICs/ FPGAs sein.

Die Wortlänge ist 32 Bit (Bild 10), die durch 4 Bit Pausen getrennt sind. Der Empfänger ist im „Label“ codiert. Die Felder SDI („Source/Destination Identifier“) und SSM („Sign/Status Matrix“) sind optional. SDI ist eine Erweiterung des Labels. Das Parity Bit ist ungerade.



Typisches IC war der Harris HS-3282 mit 2 Empfängern und 1 Sender (Bild 9). Die beiden Empfänger benötigten keine externen Interface-ICs. Der Sender hat eine FIFO für acht 32 Bit Worte. Die Empfänger konnte zumindest den Vergleich auf einen Referenzwert für die beiden SDI-Bits machen. Erst im verbesserten Holt HI-8584 konnte der Empfänger zusätzlich gegen 16 Referenzwert für die 8 Bit Labels vergleichen. Dort ist auch die FIFO des Senders auf 32 Worte erhöht worden. Beides entlastet den Controller, was in der 100 kHz Betriebsart wünschenswert ist.

Bild 8: Innenbeschaltung zu Bild 7

Protokoll

Auf die Darstellung dieses Teils von ARINC 429 sei hier verzichtet. Aus Gründen der Rückwärtskompatibilität etwas verwickelt. Und wohl nur in Zusammenhang mit konkreten Gerätekonfigurationen in Flugzeugen verständlich.

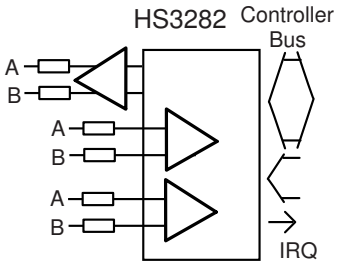


Bild 9:UART

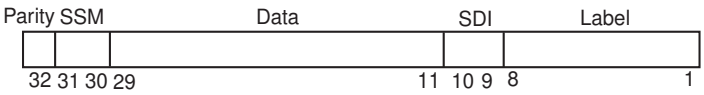


Bild 10: Datenwort

Resüme

Obwohl es für ARINC 429 also (vermutlich teuere) ICs gibt, ist rein wegen des hohen Verkabelungsaufwands das Preis/ Leistungsverhältnis ungünstig. Anzumerken ist allerdings, daß es in solchen simplen Systemen keine Kollisionen durch Zugriff

mehrerer Sender gibt. Die zeitliche worst-case Reaktionsfähigkeit ist also sicher abschätzbar.

[1] www.arinc.com
[2] Braiman, "An ARINC 429 Commentary",
SBS Technologies 1988, www.sbs-avionics.com
[3] Decker, "Avionics Data Bus Basics"
Data Device Corporation
[4] Holt Integrated Circuits
www.holtic.com
[5] Device Engineering Inc.
www.deiaz.com
[6] Data Device Corporation
www.ddc-web.com
[7] Intersil (Harris)
[8] Fairchild (Raytheon)
[9] Furse, Haupt "Down to the Wire, The Hazard of Aging Aircraft Wiring", IEEE Spectrum 2/2001

Neuigkeiten vom Mikrocontrollerverleih

Im Mikrocontrollerverleih der Forthgesellschaft hat es in den letzten Wochen einige Veränderung gegeben. Thomas Prinz, der sich in den letzten Jahren um den Verleih gekümmert hat, muss diese Aufgabe aus beruflichen Gründen abgeben. Ich (***Carsten Strotmann***) habe mich bereit erklärt, diese Aufgabe zu übernehmen.

Um den Verleih attraktiver zu machen, wird es bis zur Jahrestagung eine Webseite mit Bildern, Informationen und Verfügbarkeit unter der bekannten Adresse "**http://www.forth-ev.de/**" geben. Weitere Verbesserungsvorschläge sind gerne willkommen. Ausgewählte oder neue Mikrocontroller werde ich zukünftig in der VD vorstellen.

Ab dem 1.1.2005 gelten für den Mikrocontrollerverleih diese Ausleihregeln:

* die Kits aus dem Verleih werden für 3 Monate entliehen
* die Verleihfrist kann 3 mal um je 1 Monat verlängert werden, sofern keine Vorbestellung für das entliehene Kit vorliegt

Ausnahmeregelungen sind in begründeten Fällen möglich. Ziel dieser Regelung ist es, das keine Kits irgendwo "verstauben", wenn diese in anderen Projekten benötigt werden.

Zu den entliehenen Kits wird die Information erbeten (freiwillig), zu welchem Zweck das Kit benutzt wird. Ziel ist es, ähnliche Projekte innerhalb der Forthgesellschaft besser zu vernetzen und doppelte Arbeit zu vermeiden.

Weitere Informationen, z.B. über µCore-Kits, können Sie der nächsten Ausgabe der VD entnehmen.

Carsten Strotmann
mailto:mikrocontrollerverleih@forth-ev.de
mailto:mcv@forth-ev.de

Lineare Interpolation in Tabellen

Rafael Deliano

Ein simples, wohlbekanntes Verfahren wird genauer, wenn man die Tabellen optimiert.

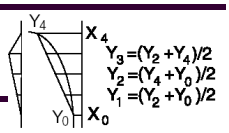
Hier anhand der Berechnung **y = x ^ 2,5** dargestellt. Wobei für x Werte von 0 ... 400h berechnet werden sollen. Eine direkte 32-Bit-Tabelle würde 4 kByte Speicher benötigen. Die hier gewählte Alternative ist, eine verkürzte Tabelle zu verwenden, die nur jede 4. Stützstelle enthält und damit nur 1 kByte belegt. Die Berechnung der 3 anderen Punkte durch lineare Interpolation (Bild 1) erfordert nur Additionen und Shiftbefehle und ist damit auch auf Controllern sehr leicht durchführbar. Dabei ist eine Sprungzieltabelle, die von den untersten beiden Bits von x gesteuert wird, eine geeignete Art der Implementierung (Listing LINT).

Optimierung

Wie in Bild 1 auch dargestellt ist, sollte man nicht einfach die ausgedünnten Werte der 4-kByte-Tabelle verwenden. Man hätte dann zwar an den Stützstellen einen minimalen Fehler, aber an den interpolierten Punkten um so mehr Abweichung. Wünschenswert ist eine gleichmäßig kleine Abweichung an allen Punkten. Wie in Bild 1 dargestellt, dadurch erreichbar, daß man den Wert an den Stützstellen etwas verschiebt.

Diese Umrechnung kann man durch eine passende Routine automatisch vornehmen: Maximalen Fehler an der Stützstelle und an den 3 davor liegenden Punkten berechnen und abspeichern. Dann den Wert an der Stützstelle dekrementieren bzw. inkrementieren (je nach Vorzeichen der Abweichung an den Stützstellen). Darauf nochmal den maximalen Fehler an allen 4 Punkten berechnen. Ist er größer geworden als der gespeicherte Wert, den letzten Schritt rückgängig machen und die Optimierung beenden. Als alternative Abbruchbedingung wird geprüft, ob das Vorzeichen des Fehlers wechselt. Das heißt, ob man den Wert an der Stützstelle so weit verschoben hat, daß er selbst nun den maximalen Fehler enthält, was natürlich nicht sein soll.

Man sollte sich anhand der 4-kByte-Tabelle als Referenz den Fehler der berechneten Punkte ausdrucken (Tabelle 1). Bei der Funktion **x ^ 2,5** ist der relative Fehler anfangs extrem hoch. Am Ende der Tabelle wird er geringer. In der beabsichtigten Anwendung waren Werte unterhalb 50h und oberhalb 200h von geringer Bedeutung, so daß keine weiteren Optimierungen nötig waren.




```
Tabelle 1: Fehler
unoptimierte Tabelle:          optimierte Tabelle:
             Ref   Fehler =                       Ref   Fehler =
   X           Y   Approx          X            Y   Approx
                   - Ref                          - Ref
0000   0000 0000   0000 <-    0000   0000 0000 + 0000 <-
0001   0000 0001   0007        0001   0000 0001 + 0005
0002   0000 0006   000A        0002   0000 0006 + 0007
0003   0000 0010   0008        0003   0000 0010 + 0003
0004   0000 0020   0000 <-    0004   0000 0020 - 0006 <-
0005   0000 0038   000D        0005   0000 0038 + 0006
0006   0000 0058   0012        0006   0000 0058 + 000A
0007   0000 0082   000D        0007   0000 0082 + 0004
0008   0000 00B5   0000 <-    0008   0000 00B5 - 000A <-
...
03F8   01F6 0EFB   0000 <-    03F8   01F6 0EFB - 0077 <-
03F9   01F7 4B79   00B3        03F9   01F7 4B79 + 003C
03FA   01F8 886E   00EF        03FA   01F8 886E + 0078
03FB   01F9 C5DB   00B3        03FB   01F9 C5DB + 003C
03FC   01FB 03BF   0000 <-    03FC   01FB 03BF - 0077 <-
03FD   01FC 421C   00B3        03FD   01FC 421C + 003C
03FE   01FD 80F0   00EF        03FE   01FD 80F0 + 0078
03FF   01FE C03C   00B3        03FF   01FE C03C + 003C
0400   0200 0000   0000 <-    0400   0200 0000 - 0078 <-
```

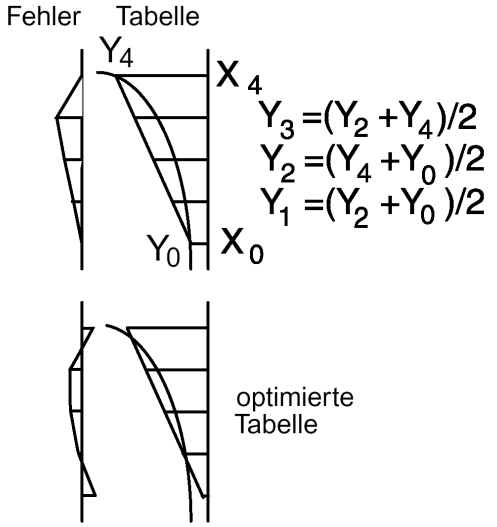


Bild 1: Interpolation, direkt & optimiert

```
<| \ LINT
HEX
TABLE 2,5-TAB
0000 0000 D, 001A 0000 D,
00AB 0000 D, 01E7 0000 D,
...
3B61 01EC D, 2138 01F1 D,
0E84 01F6 D, 0348 01FB D,
FF88 01FF D,

: 2,5-00    \ ( UN1 — UD1 )
2,5-TAB + D@ ;

: 2,5-01    \ ( UN1 — UD1 )
FFFC AND 2,5-TAB + DUP D@
ROT 4 + D@ 2OVER D+ UD2/
D+ UD2/ ;

: 2,5-10    \ ( UN1 — UD1 )
FFFC AND 2,5-TAB + DUP D@
ROT 4 + D@ D+ UD2/ ;

: 2,5-11    \ ( UN1 — UD1 )
FFFC AND 2,5-TAB + DUP 4 +
D@ ROT D@ 2OVER D+ UD2/
D+ UD2/ ;

TABLE 2,5-JTAB
' 2,5-00 , ' 2,5-01 ,
' 2,5-10 , ' 2,5-11 ,

: x^2,5  \ ( UN1 — UD1 )
          \UN1 = 0 ... 400h
DUP 3 AND 1<SHIFT 2,5-JTAB
 + @ EXECUTE ;
|>
```

Reisen bildet, aber fragen Sie Bill Gates besser nicht nach dem Weg!

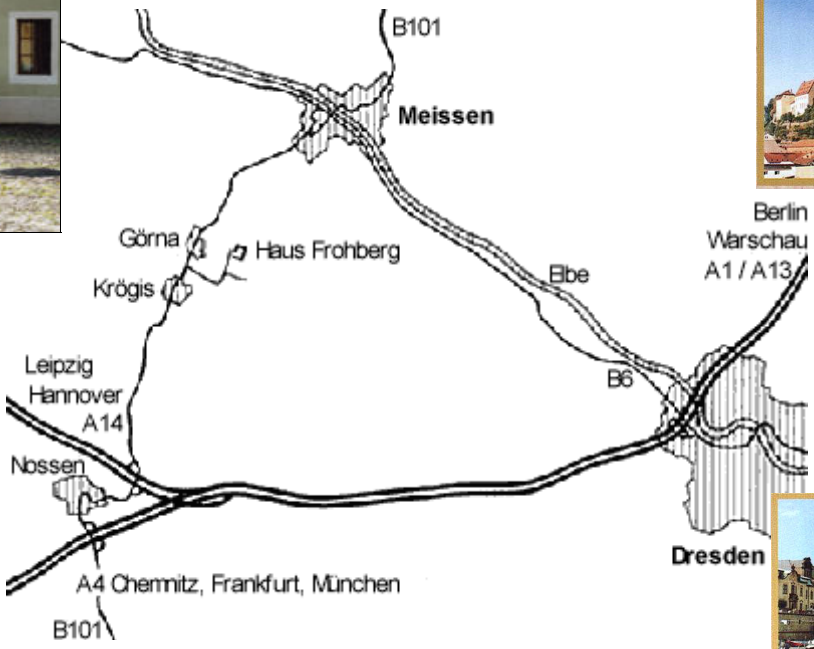
Schauen Sie einmal bei

http://mappoint.msn.com/DE/DirectionsFind.aspx

vorbei.

Start Adresse: Norwegen / Ort: Trondheim
Ziel Adresse: Norwegen / Ort: Haugesund

Route abfragen: Viel Spaß ;-)



(7.) 8. bis 10. April 2005

Landhotel Gut Frohberg
OT Schönnewitz 9
01665 Krögis b. Meissen

Forth-Gruppen regional

Moers

Friederich Prinz
Tel.: (0 28 41) - 5 83 98 (p) (Q)
(Bitte den Anrufbeantworter nutzen!)
(Besucher: Bitte anmelden!)
Treffen: 2. und 4. Samstag im Monat
14:00 Uhr, **MALZ, Donaustraße 1**
47441 Moers

Mannheim

Thomas Prinz
Tel.: (0 62 71) - 28 30 (p)
Ewald Rieger
Tel.: (0 62 39) - 92 01 85 (p)
Treffen: jeden 1. Mittwoch im Monat
Vereinslokal Segelverein Mannheim e.V.
Flugplatz Mannheim-Neuostheim

München

Jens Wilke
Tel.: (0 89) - 8 97 68 90
Treffen: jeden 4. Mittwoch im Monat
Ristorante Pizzeria Gran Sasso
Ebenauer Str. 1
80637 München

Hamburg

Küstenforth
Klaus Schleisiek
Tel.: **(0 40) - 37 50 08 03 (g)**
kschleisiek@send.de
Treffen 1 Mal im Quartal
Ort und Zeit nach Vereinbarung
(bitte erfragen)

Gruppengründungen, Kontakte

Hier könnte Ihre Adresse oder Ihre Rufnummer stehen – wenn Sie eine Forthgruppe gründen wollen.

µP-Controller Verleih

Carsten Strotmann
mikrocontrollerverleih@forth-ev.de
mcv@forth-ev.de

Forth-Hilfe für Ratsuchende

Jörg Plewe
Tel.: **(02 08) - 49 70 68** (p)

Spezielle Fachgebiete

FORTHchips
(FRP 1600, RTX, Novix)

Klaus Schleisiek-Kern
Tel.: **(0 40) - 37 50 08 03** (g)

KI, Object Oriented Forth,
Sicherheitskritische
Systeme

Ulrich Hoffmann
Tel.: **(0 43 51) - 71 22 17** (p)
Fax: **- 71 22 16**

Forth-Vertrieb
vlksFORTH
ultraFORTH
RTX / FG / Super8
KK-FORTH

Ingenieurbüro **Klaus Kohl**
Tel.: **(0 82 33) - 3 05 24** (p)
Fax :**(0 82 33) - 99 71**
forth@designin.de

Möchten Sie gerne in Ihrer Umgebung eine lokale Forthgruppe gründen, oder einfach nur regelmäßige Treffen initiieren? Oder können Sie sich vorstellen, ratsuchenden Forthern zu Forth (oder anderen Themen) Hilfestellung zu leisten? Möchten Sie gerne Kontakte knüpfen, die über die VD und das jährliche Mitgliedertreffen hinausgehen?

Schreiben Sie einfach der VD - oder rufen Sie an - oder schicken Sie uns eine E-Mail!

Hinweise zu den Angaben nach den Telefonnummern:

Q = Anrufbeantworter
p = privat, außerhalb typischer Arbeitszeiten
g = geschäftlich

Die Adressen des Büros der Forthgesellschaft und der VD finden Sie im Impressum des Heftes.

Business, industry, and education are discovering that Forth is an especially effective language for producing compact, efficient applications for real-time, real-world tasks. And now there's **Thinking Forth**—an instructive guide that illustrates the *elegant logic* behind the language and shows you how to apply specific problem-solving tools to software, regardless of your programming environment.

It combines the philosophy behind Forth with traditional, disciplined approaches to software development—to give you a basis for writing more readable, easier-to-write, and easier-to-maintain software applications in any language.

Written in the same lucid, humorous style as the author's *Starting Forth* and packed with detailed coding samples and illustrative cartoons, **Thinking Forth** reviews fundamental Forth concepts and takes you from the initial specification of your software project through the analysis and implementation process, showing how to simplify your program and still keep it flexible throughout. Both beginning and experienced programmers will gain a better understanding and mastery of such topics as

- Forth style and conventions
- decomposition
- factoring
- handling data
- simplifying control structures
- and more.

And, to give you an idea of how these concepts can be applied, **Thinking Forth** contains revealing interviews with real-life users and with Forth's creator, CHARLES H. MOORE.

To program intelligently, you must first *think* intelligently, and that's where **Thinking Forth** comes in.

Leo Brodie is a writer, programmer, consultant, teacher, and world-renowned authority on Forth. He was a technical writer for Forth Inc., and has been an independent consultant (since 1982) for such clients as IBM, NCR, and Lockheed. He is also the author of *Starting Forth* (Prentice-Hall, 1981).

ISBN 0-13-917568-7
90000
9 780139 175688

THINKING FORTH

LEO BRODIE

Leo Brodie

THINKING FORTH

A Language and Philosophy for Solving Problems

Includes interviews with Forth's inventor, CHARLES H. MOORE, and other Forth thinkers

1 / 2005