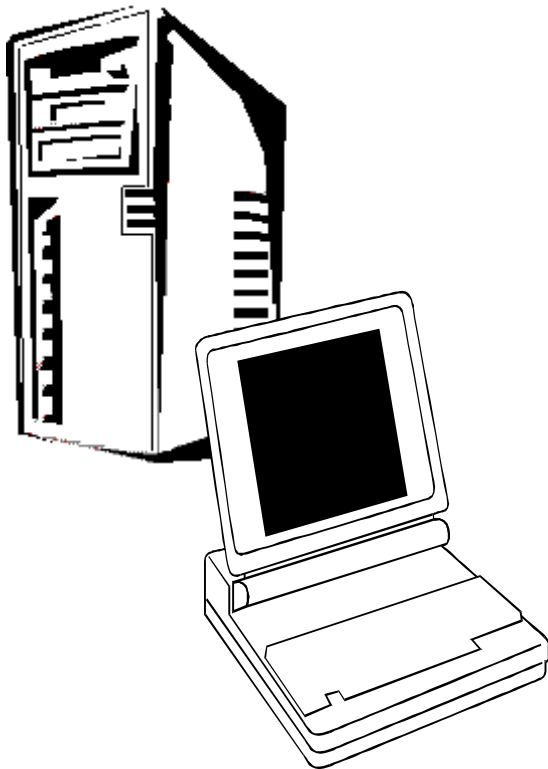
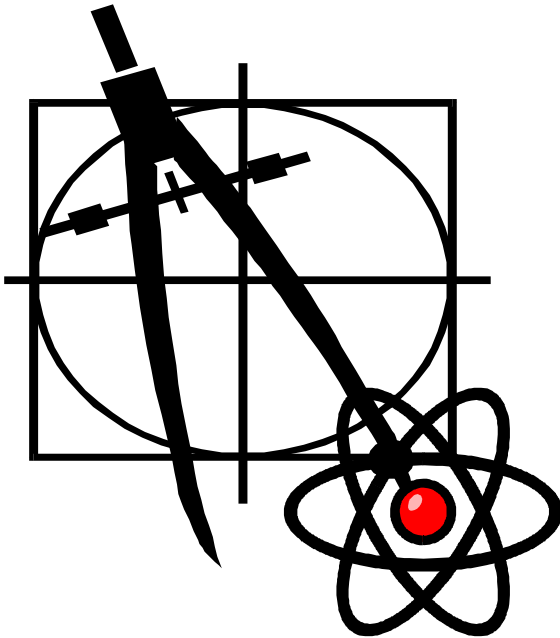


für Wissenschaft und Technik, für kommerzielle EDV,
für MSR-Technik, für den interessierten Hobbyisten.



In dieser Ausgabe:

Leserbriefe und Rezensionen

Leser schreiben, was sie interessiert

Codeknacker – 2

Fortsetzung des Aufsatzes über polyalphabetische
Verschiebungen

Farbe ins Leben

Druckeransteuerung mit Win32FOR

Windows2000

Erste Erfahrungen eines voreiligen Nutzers

Neue Rätsel

An diesen Nüssen haben Sie zu knacken

DO...LOOP über 32 kByte hinweg

Erweiterungen für ZF und TurboForth

Dienstleistungen und Produkte fördernder Mitglieder des Vereins

FORTH - Shirt



Räumungsverkauf

T - Shirt: hellgrau / grün
in Größe M-L-XL **15 DM**

Sweat-Shirt: grau / grün
in Größe M-L-XL **25 DM**
(+ Porto)

ForthWORKS

Ulrike Schnitter
Nelkenstr. 52
85716 Unterschleißheim
fon/fax 089-310 33 85

Hier könnte IHRE Anzeige stehen

Setzen Sie sich doch einfach einmal mit dem
Büro der Forthgesellschaft e.V. in Verbindung.

Dipl.-Ing. Arndt Klingenberg

Tel.: ++32 +87 -63 09 89 (Fax: -63 09 88)
Waldring 23, B-4730 Hauset, Belgien
akg@aachen.kbbs.org

Computergestützte Meßtechnik und Qualitätskontrolle, Fuzzy, Datalogger, Elektroakustik (HiFi), MusiCassette High-SpeedDuplicating, Tonband, (engl.) Dokumentationen und Bedienungsanleitungen

Forth Engineering Dr. Wolf Wejgaard

Tel.: +41 41 377 3774 - Fax: +41 41 377 4774
Neuhöflirain 10
CH-6045 Meggen <http://holonforth.com>

Wir konzentrieren uns auf Forschung und Weiterentwicklung des Forth-Prinzips und offerieren HolonForth, ein interaktives Forth Cross-Entwicklungssystem mit ungewöhnlichen Eigenschaften. HolonForth ist erhältlich für 80x86, 68HC11 und 68300 Zielprozessoren.

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

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.

FORTECH Software Entwicklungsbüro Dr.-Ing. Egmont Woitzel

Joachim-Jungius-Straße 9 D-18059 Rostock
Tel.: (0381) 405 94 72 Fax: (0381) 405 94 71

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 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.

Impressum	4
Editorial	4
Leserbriefe	5
Was Sie uns und den Lesern der Vierten Dimension mitteilen wollten	
Codeknacker, Teil 2	7
Fortsetzung des Aufsatzes über polyalphabetische Verschiebungen, <i>Hugh Aguilar</i>	
Farbe ins Leben	12
Druckeransprache mit Win32For, <i>Martin Bitter</i>	
Gehaltvolles	13
Rezensionen div. Ausg. der Fourth Dimension, Forthwrite und Feigenblatt, <i>Fred Behringer</i>	
Windows2000	22
Erste Erfahrungen, <i>Friederich Prinz</i>	
Der verschlüsselte Schlüssel oder X S 2XOR	24
Kniffliges, <i>Fred Behringer</i>	
Rätsel	25
Kniffligeres, <i>Fred Behringer</i>	
Neues aus der FIG Silicon Valley	26
<i>Henry Vinerts</i>	
Wanted	28
Eine Rubrik soll wieder aufleben, <i>Fred Behringer</i>	
DO--LOOP mit Code von bis zu 32 K Länge	29
ab 80386 für Assembler in ZF und Turbo-Forth, <i>Fred Behringer</i>	

In der nächsten Ausgabe finden Sie voraussichtlich:

- was immer SIE uns zur Veröffentlichung zuschicken !

IMPRESSUM

Name der Zeitschrift

Vierte Dimension

Herausgeberin

Forth-Gesellschaft e.V.
Postfach 16 12 04
D-18025 Rostock
Tel.: 0381-400 78 28

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
Homburgerstraße 335
47443 Moers
Tel./Fax.: 02841-58 3 98
E-Mail:
VD@FORTH-EV.DE
FRIEDERICH.PRINZ@T-ONLINE.DE

Anzeigenverwaltung

Büro der Herausgeberin

Redaktionsschluß 1999

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

Erscheinungsweise

1 Ausgabe / Quartal

Einzelpreis

DM 10,- zzgl. Porto u. Verp.

Manuskripte und Rechte

Berücksichtigt werden alle eingesandten Manuskripte. Leserbriefe können ohne Rücksprache gekürzt wiedergegeben werden. Für die mit dem Namen des Verfassers gekennzeichneten Beiträge übernimmt die Redaktion lediglich die presserechtliche Verantwortung. Die in diesem Magazin veröffentlichten Beiträge sind urheberrechtlich geschützt. Übersetzung, Vervielfältigung, Nachdruck sowie Speicherung auf beliebigen Medien ist auszugswise nur mit genauer Quellenangabe erlaubt. Die eingereichten Beiträge müssen frei von Ansprüchen Dritter sein. Veröffentlichte Programme gehen - soweit nichts anderes vermerkt ist - in die Public Domain über. Für Fehler im Text, in Schaltbildern, Aufbauskizzen u.ä., die zum Nichtfunktionieren oder eventuellem Schadhafwerden von Bauelementen oder Geräten führen, kann keine Haftung übernommen werden. Sämtliche Veröffentlichungen erfolgen ohne Berücksichtigung eines eventuellen Patentschutzes. Warennamen werden ohne Gewährleistung einer freien Verwendung benutzt.



Liebe Leser,

zunächst hoffe ich, daß Sie diese Ausgabe unserer Quartalsschrift noch vor dem Jahrestreffen der Forthgesellschaft e.V. in den Händen halten. In diesem Fall möchte ich Sie daran erinnern, daß nicht mehr viel Zeit bis zu unserem Treffen in Hamburg bleibt ! Bitte melden Sie sich an, kommen Sie nach

Hamburg und genießen Sie mit uns ein Wochenende unter Forthern !

Einen zweiten, dringenden Aufruf möchte ich an diese Stelle plazieren. Schreiben Sie doch wieder einmal ein paar Seiten für unsere VD ! Die allermeisten von Ihnen arbeiten an vielen großen und weniger großen Projekten, an interessanten Dingen über die es ganz sicher immer wieder einmal ein paar Zeilen zu berichten gibt. Berichtenswert sind diese Arbeiten allemal, und interessant sind sie für die Leser der VD ganz sicher ebenfalls. Bitte nehmen Sie sich die Zeit und schreiben Sie uns – damit IHRE VD schon ab der nächsten Ausgabe wieder 36 oder sogar 40 Seiten mächtig wird.

Ich möchte Ihre Aufmerksamkeit auf Henry Vinert's Brief aus der FIG Silicon Valley lenken. Der darin beschriebene Schwund an Mitgliedern ist uns hier nicht Neues. Neu scheint mir die resignative Grundstimmung zu sein, die in den USA offenbar nicht nur den Henry erfaßt hat. Auch Charles Moore sieht in Forth nicht den ‚Weg der Zukunft‘. Ich denke, daß Henry's Brief sicher eine Menge Diskussionsstoff bietet und Anlaß für viele Leserbriefe sein sollte.

Und auch die neuen ‚Rätsel‘ werden Sie vermutlich eine Weile beschäftigen. Ihre Lösungen erwarten wir hier natürlich ‚händeringend‘. Interessanter als die Lösungen selbst, sind vielleicht die unterschiedlichen Ansätze, mit denen Sie zu Ihren Lösungen gelangt sind.

Was mich persönlich in diesen Tagen besonders beschäftigt, ist das Windows 2000. Es ist endlich (leider ?) auch in einer deutschsprachigen Version verfügbar. Sie können in diesem Heft meinem ersten ‚Erfahrungsbericht‘ entnehmen, daß wieder auf Microsoft geschimpft werden darf. Sie wissen, daß ich nichts davon halte, selbstverschuldete Fehler (meist im Speichermanagement) dem Windows ‚in die Schuhe zu schieben‘. Aber Windows2000 hält zur Zeit wenig von dem, was Microsoft vollmundig (wie immer) versprochen hat. Wie sind Ihre Erfahrungen mit dem neuen System und den alten Fehlern ?

Wenn man Presse, Funk und Fernsehen Glauben schenkt (wer tut das noch ?), kann man den Eindruck gewinnen, daß die ganze werktätige (im Büro) Bevölkerung mit großem persönlichen Einsatz daran arbeitet, täglich Millionen von teuren Arbeitsstunden mit der Jagd auf Moorhühner zu verschwenden. Der deutschen Wirtschaft entstehen täglich Schäden in Millionenhöhe – allerdings bisher ohne nachweisbare Auswirkungen auf die Insolvenzstatistiken. Spielen Sie mit ? Jagen Sie auch Moorhühner ? Ich tu's, selbstverständlich nur zu Hause, und nur dann, wenn die Arbeit an der VD mir dazu Zeit läßt. Und ich kann an der Moorhuhnjagd nur einen einzigen, ärgerlichen Punkt entdecken: Das Spiel ist nicht mit Forth geschrieben.

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



Betreff: meine falsche eMehl Adr - al capo
 Von: akg@aachen.forth-ev.de (Arndt Klingelberg)
 Ruckantwort: akg@aachen.kbbs.org

Dear Vd ,
 habe mal wieder in meine Adresse vorne UND hinten reingelesen. Das habe ich zwar schon X-mal diskutiert aber meine e-Mehl adresse ist immer noch falsch, immerhin aber vorne und hinten gleich.
 Die emehl Adresse steht im vor und/oder Nachspann dieser Mehl, aber BITTE die richtige nehmen, die 'reply to'

Wenn ihr dann die Adresse mal richtig habt, stelle ich meine Adresse um.

Danke

ps. = knappes kW
 wie wäre es eure E-Mehl adress in 4D@.... zu doppeln wegen der venereal disease

virtually
 Arndt Klingelberg 8-)

*Dear Arndt,
 ...wie es Euch gefällt...Allerdings habe ich eine Mail an Dich, gerichtet an die ‚reply to‘ Adresse, mit dem Vermerk zurückbekommen, daß diese Adresse unbekannt sei. Gleiches gilt, leider, auch für die Aachener Adresse. Kann es sein, daß Du dieses mal Deine Adresse bereits vorher umgestellt hast ? ;-)
 Really*

Friederich Prinz

P.S. Das mit der venereal disease ist bekannt, ja und ? ;-)

Betreff: Leserbrief
 Von: Fred Behringer <behringe@sunstatistik1.mathematik.tu-muenchen.de>
 Ruckantwort: behringe@mathematik.tu-muenchen.de

Lieber Fritz,
 eine Zeitschrift, in der man alles versteht und besser weiß, gefällt einem weniger gut. Wenn man gar nichts versteht und die Achtung vor den Autoren ins Unermeßliche steigt, gefällt sie einem schon besser. Am besten gefällt sie einem, wenn man ein paar Dinge besser weiß, einiges gerade mal so versteht und einiges neu ist und zu weiteren Studien anregt.
 Die VD hat mir diesmal wieder ganz besonders gut gefallen. Ich werde mich gleich daranmachen, für Alan Wenham eine Übersicht zusammenzustellen, damit er uns in der Forthwrite gebührend darstellen kann.
 Bewundernswert finde ich die große Mühe, die sich Deine Frau mit dem Korrekturlesen macht. Mir fällt sowas sofort und sehr angenehm auf. Die Lesbarkeit steigt mit jedem einzelnen Komma und man braucht beim sequentiellen Lesen, auf das man in der geschriebenen Sprache nun mal angewie-

sen ist, nicht bei jedem nur unbestimmt fortsetzbaren Satzteil neu anzusetzen. Man müßte Deiner Frau wirklich einen Sonderorden verleihen.

Die Schmunzelgeschichte von Sören Tiedemann ist nicht schlecht. Allerdings glaube ich kaum, daß sich Mathematiker mit einem Elefanten abgeben, es sei denn, man erklärt letzteren per definitionem zum mathematischen Begriff. Das ist ohne Beschränkung der Allgemeinheit trivial, würde Albrecht Beutelspacher mit dem Titel seines Buches sagen.

Gefallen hat mir auch der Artikel von Wolfgang Allinger über "CFA2NAME". Ein "substantieller" Beitrag, mit dem man wirklich etwas anfangen kann, nicht nur in F-PC. Zuerst dachte ich: "da gibt es doch >NAME, was ja nichts anderes tut, als zur cfa die zugehörige nfa zu liefern". Aber dann wurde mir klar, daß das Problem ganz woanders steckt. Eine Art "inverses" Problem.

Und schließlich hat Deine Artikelfortsetzung "Hashing" auf mich Eindruck gemacht. So sehen Artikel aus, aus denen man Anregung für eigene Arbeiten schöpfen kann. "Das langweilt den Freak", argwöhnst Du. Mitnichten. Freak oder nicht Freak, das muß ein armer Forthler sein, der sich nicht für seine Werkzeuge interessiert. Man kann nicht oft genug nachlesen, wie man trotz HASH ein ganzes Vokabular durchforsten kann (WORDS), und, falls nötig, über VOC-LINK ganze Vokabulare (für die Systemstatistik beispielsweise beim Metacompilieren). Und dann die (nicht nur mit Vorteilen verbundene) Trennung zwischen Headern und Bodies! Deine Bemerkung über YDUMP ist gut. Mit DUMP-32 aus VD 2/98 ginge es in einem einzigen Rutsch! Und per Knopfdruck durchscrollbar! Vorwärts und rückwärts! Bei mir über eine Strecke von 256 Megabyte RAM!

Was man nur noch schnell bräuchte, wäre ein System, durch das das DUMP-32 256 Megabyte weit hindurchdumps könnte.

Herzlichen Gruß

Fred (Behringer)

Betreff: 4D & the DAO of forth & der Sinn des Lebens
 Von: Martin.Bitter@t-online.de (Martin Bitter)

Hallo Fritz,
 ich habe dem Henry den folgenden _CODE_ gemailt, mit der Bitte, Kontakt zur Forth Dimensions herzustellen. Der Begleittext ist natürlich ein anderer. Vielleicht wäre das was für die 4D? (Es sind zwei unabhängige Beiträge)

Nabel oder der Sinn des Lebens

In der letzten 4D war die ausgezeichnete Übersetzung eines Artikels von H.C. Ting zu lesen. "The Dao of Forth" (Der Forthweg). Ich habe einige Zeit über dieses Tao meditiert und kam zu folgender Erkenntnis.

Verbindet man Siddarthas Wissen über die 'Sieben Stufen der Erleuchtung' mit C.H. Tings Tao des Forth, so könnte ein Weg so aussehen:



Leserbriefe

Vocabulary philosophie
Vocabulary östlicher_diwan also

Step 1 Betrachte das Tao des Forth bis du zwei Nabel
erkenntst.

: Dao (--) Nabel1_x Nabel1_y Punkt
Nabel2_x Nabel_2y Punkt ;

Step 2 : r (-- n) Nabel1_y Nabel2_y - ;

Step 3 : m (-- x y) Nabel1_x r 2/ + Nabel1_y ;

Step 4 : welt (--) m r circle ;

Step 5 : ying (--) Nabel1_x Nabel1_y r 2/ 0 180 arc
Nabel1_x 2 - Nabel1_y
white flood_fill ;

Step 6 : yang (--) Nabel2_x Nabel2_y r 2/ 180 0 arc
Nabel1_x 2 + Nabel1_y
black flood_fill ;

Step 7 : ZEN (--) Dao ying yang ;

Tiefe Einsichten über das Tao des Forth oder Hilfe für den
Novizen.

Nehme das 'Tao des Forth' und drehe es um 90 Grad (drehe in
die Richtung, die deiner politischen Überzeugung am näch-
sten kommt). Betrachte. Meditiere. Betrachte! Drehe deine
Augen einwärts, so das du die 'Dreifaltigkeit der Punkte'
siehst. Ein flacher, blaßer Punkt links, ein flacher, blaßer
Punkt rechts. In der Mitte ein deutlicher, irgendwie tiefen-
scharfer, schwebender Punkt!

Jetzt has du das Tao!

Trachte danach, dass dir in diesem Moment niemand in die
Augen schaut! ;-)

Martin Bitter

Betreff: Forth in der c't

Von: Martin Bitter <mbitter@bigfoot.de>

Hallo,

Forth ist ja schon lange nicht mehr in der c't erwähnt worden.
Aber in der Nummer 5 steht was auf Seite 58!

Robotor am MIT werden in FORTH programmiert!

Immerhin ;-)

Martin Bitter

Betreff: LEGO-RCX-Buch und Forth

Von: Martin Bitter <mbitter@bigfoot.de>

Bei Galileo Press ist 'Dave Baums Lego-Mindstroms Robotor'
erschienen (das ist ganz genau der Titel, da sind noch ein
paar Trademarkzeichen drin, aber die kann ich hier nicht
wiedergeben). Ich hab's noch nicht gelesen, aber in der ameri-
kanischen Version wird auch das pbForth von Ralph Hempel
erwähnt - warum sollte das in der deutschen Ausgaben anders
sein?

Das Buch wird zwar schon beworben, aber es ist (jetzt!) bei
BOL noch als nicht lieferbar geführt. Trotzdem der Preis steht
schon fest. 69 DM mit CD-Rom.

Wer Interesse hat, kann ja mal reinschauen (sobald es da ist)!
Gruß

Martin

Leserbrief

Modula-2 endgültig out

Ich habe mir heute in der TU-Bibliothek ein ausrangiertes Exemplar
des Buches "Programmieren in Modula-2" von Niklaus Wirth gesi-
chert. "Zum Mitnehmen." Sie lagen da zu zwanzig auf dem Haufen.
In der Studentenbibliothek. Meine Frage an die Bibliothekarin:
"Warum denn das?" Ihre Antwort: "Hat keiner mehr ausgeliehen."
Was ist das nur für eine kurzlebige Welt, diese unsere ??? Das heißt
aber dann doch auch, daß ganze Generationen von Informatikern -
Verzeihung, ganze Jahrgänge; die Informatik besteht ja noch gar
nicht so lange - mit Modula-2 gequält wurden, das sie jetzt, nach
dem langen Marsch durch die Institutionen, in ihren inzwischen er-
reichten Führungspositionen gar nicht mehr brauchen? Oder viel-
leicht doch? Immer noch? An veralteten Maschinen? Das einzige,
was ewigen Bestand zu haben scheint, ist Forth (?) Da kann man
über die Generationen hinweg darüber philosophieren, "daß alles
andere schon darin enthalten ist, schon längst darin enthalten war".
Wenn man aber sowieso schon dauernd umlernen muß, warum be-
schäftigt man sich dann an den Hochschulen überhaupt mit dem ER-
LERNEN von Sprachen? Warum geht man nicht einfach hin und
SPRICHT sie? FORTH lernt man nicht. Forth kann man gar nicht
lernen. Es bildet sich ständig forth. FORTH spricht man - schlecht
und recht (mehr schlecht denn recht) - oder man läßt das Program-
mieren ganz sein.

Fred Behringer

Hallo Friederich.Prinz,

ich sortiere gerade Stapel von Zeitschriften der unterschied-
lichsten Art. Vielleicht sollte das Redaktorial mal darüber
nachdenken, warum 99.99 % aller Zeitschriften die Ausgabe
(Jahrgang und Nr) OBEN rechts, oft auch zusätzlich auch
OBEN links oder OBEN auf dem Rücken führen.

Dazu noch ziemlich FETT.

Ansonsten hoffe ich, daß es euch gut geht.

Arndt Klingelberg

Hallo Arndt,

es geht uns gut, meistens, danke der Nachfrage.

Der Hinweis hat hier in Moers die unterschiedlichsten Reaktionen
ausgelöst, von „was soll DAS denn“ über „DER hat vielleicht Sor-
gen“ bis „Na klar, Recht er, ich ärgere mich jedesmal, wenn ich die
in Kartons archivierten, aufrecht stehenden VD komplett aus dem
Karton herausnehmen muß, um Jahrgang und Ausgabe lesen zu kön-
nen.“

ok – wir haben verstanden !

Ausgabe und Jahrgang sind jetzt oben rechts, bzw. oben links so-
wohl auf der Vorder- als auch auf der Rückseite zu finden; fett und
deutlich genug, hoffen wir.

fep

P.S. ...hätte man uns auch schon mal früher sagen können ;-)



Ein Programm zum Knacken von polyalphabetischen Codes (Teil 2)

Hugh Aguilar <haguilar@forth.org>, Phoenix, USA

Mit freundlicher Genehmigung der amerikanischen Forth Interest
Group übersetzt von Fred Behringer, München.
Der Originalaufsatz erschien in der Forth Dimensions, Band XX,
Nummer 5, 6 vom Januar/April 1999.

Das ist der zweite Teil unseres Artikels über das Knacken der polyalphabetischen Substitutionsverschlüsselung (der PolySub). Im ersten Teil hatten wir dargestellt, was der PolySub ist, und ein Programm zur Ermittlung der Schlüssellänge besprochen. Die Schlüssellänge muß uns bekannt sein, bevor wir an die Ermittlung des Schlüsselinhalts gehen können. Im vorliegenden Teil werden wir uns an die Ermittlung des Schlüsselinhalts machen. Das Programm erfordert interaktive Mithilfe von seiten des Benutzers, besonders wenn der Versuch unternommen wird, kleine Dateien, die mit langen Schlüsseln verschlüsselt sind, zu entziffern.

Phase 2 --- Ermittlung des Schlüsselinhalts

Nun sind wir für Phase 2 bereit: Wir wollen den Schlüsselinhalt herausbekommen. Nach den einzelnen Zeichen des Schlüssels wird so gesucht, als ob es sich um voneinander verschiedene monoalphabetische Schlüssel handelte. Wir hatten das Gefühl, daß die zweite Phase des Programms direkter auf ihr Ziel zusteuert als die erste. Von nun an geht es zügig voran!

In Screen 1 befindet sich eine Variable namens **MOSTEST**, die dasjenige Klartextzeichen enthält, das wir für das häufigste halten. Vorgabe ist der Zwischenraum. Dieser Wert wird normalerweise während des Programmablaufs nicht verändert. **MOSTEST** wurde aber als Variable und nicht als Konstante ausgelegt, da der Anwender vielleicht Veränderungen vornehmen möchte, wenn er eine Datei entschlüsselt, die keinen Text enthält. Eine solche Veränderung kann vorgenommen werden, ohne das Programm neu compilieren zu müssen. Anzumerken ist, daß selbst in englischen Texten manchmal der Zwischenraum nicht das häufigste Zeichen ist. Man betrachte Abbildung 3, wo "e" am häufigsten vorkommt.

Auch solche Geheimtexte wird das Programm mit Erfolg knacken. Die Textdatei des englischen Originals des vorliegenden Artikels hat 1,74 mal soviele Zwischenräume wie "e"-Zeichen. Für andere Sprachen als Englisch, oder rein zufällig in kurzen Dateien, wird das Verhältnis vielleicht näher bei 1,0 liegen. Das Zeichen in **MOSTEST** braucht nicht unbedingt das wirklich *häufigste* Zeichen zu sein, solange es zu-

mindest sehr häufig ist. In unserem Datenfeld **KEY_CHAR** berechnen wir nämlich für jedes Zeichen im Schlüssel die 256 besten Ansätze. Es stehen uns verschiedene Wege offen, die "besten" Ansätze herauszufiltern, wenn sie Zeichen darstellen, die der Chiffrierer kaum in seinem Schlüssel verwendet haben dürfte.

Wir haben ein zweidimensionales Datenfeld namens **KEY_CHAR** vorgesehen, das wir füllen wollen. Die Zeile 0 von **KEY_CHAR** enthält den besten Ansatz für das, was wir für den Schlüssel halten. Zeile 1 enthält den zweitbesten Ansatz usw. Wir wollen zuerst einen Blick auf **FILL_KEY** in Screen 28 werfen und dann die Routinen auf niedrigeren Ebenen zurückverfolgen.

FILL_KEY ruft für jedes einzelne Zeichen des Schlüssels (für jede Spalte von **KEY_CHAR**) das Wort **FILL_FREQS** in Screen 26 auf. **FILL_FREQS** nimmt einen Zeiger auf **CIPHERTEXT** auf und setzt diesen um die Schlüssellänge weiter. **FILL_FREQS** zählt ab, wieviele von den einzelnen Zeichen in **CIPHERTEXT** vertreten sind. **FILL_FREQS** nimmt diese Zählung so vor, als ob es sich um eine monoalphabetische Substitutionsverschlüsselung handelt, bei der die Zeichen zufälligerweise über die einzelnen Abschnitte der Länge **KEY_LENGTH** in **CIPHERTEXT** gleichmäßig verteilt sind.

FILL_KEY ruft dann das Wort **COLUMN_FILL_KEY** auf, das genau eine Spalte von **KEY_CHAR** füllt. **COLUMN_FILL_KEY** ruft für jede Zeile **SINGLE_FILL_KEY** in Screen 27 auf. **SINGLE_FILL_KEY** nimmt die horizontalen und vertikalen Indizes auf, die es dann in **KEY_CHAR** setzt. **SINGLE_FILL_KEY** ermittelt dasjenige Geheimtextzeichen in **FREQS**, das am häufigsten auftritt, und nimmt an, daß das zu dem Klartextzeichen in **MOSTEST** gehört. **SINGLE_FILL_KEY** rechnet aus, welches Zeichen im Schlüssel dieses Geheimtextzeichen erzeugt hätte, wenn man annimmt, daß das Zeichen in **MOSTEST** das Klartextzeichen war. Das so ermittelte Zeichen wird in **KEY_CHAR** abgespeichert. **SINGLE_FILL_KEY** gibt den Index nach **FREQS** aus, der zu dem als häufigst erkannten Geheimtextzeichen gehört. **COLUMN_FILL_KEY** setzt an die betreffende Stelle in **FREQS** den Wert -1 ein, bevor es zur Berechnung des nächstwahrscheinlichen Zeichens übergeht, und zwar deshalb, damit **SINGLE_FILL_KEY** nicht immer wieder denselben Wert als Bestwert ermittelt.

Screen 30 enthält die Routine **TO_KEY_STRING**. Wir hatten ursprünglich lediglich Zeile 0 von **KEY_CHAR** nach **KEY_STRING** kopiert. Da mußte aber noch mehr gemacht werden. Wir hatten uns nicht überlegt, daß die allerwenigsten Leute einen Chiffrierschlüssel verwenden werden, der nicht-darstellbare Zeichen enthält. Wir wollen letztere herausfiltern. Es bieten sich verschiedene Wege an, Zeichen, die wir nicht haben wollen, herauszufiltern. **TO_KEY_STRING** nimmt die cfa eines char_kind-Wortes (eines aus **PRINTA-**



BLE, **NUMERIC**, **GERMAN**, **UPPERCASE**, **LOWERCASE**, **ALPHA**, **ALPHA-NUMERIC** und **PUNCTUATION**) auf. **TO_KEY_STRING** sucht jede Spalte von **KEY_CHAR** von oben nach unten ab, um das erste Zeichen der **char_kind**-Klasse zu finden, von welcher **TO_KEY_STRING** ausgegangen war. Jede Spalte von **KEY_CHAR** enthält alle möglichen Zeichen (jede Spalte hat 256 Komponenten), so daß wir also notwendigerweise etwas finden müssen, das unsere **char_kind**-Bedingung erfüllt. Auf diese Weise erhalten wir die besten Ansätze aus irgendeiner der **char_kind**-Klassen.

FILL_KEY_STRING erledigt alles, was nötig ist, um den Schlüsselinhalt zu ermitteln. **FILL_KEY_STRING** verwendet **ALPHA** als Vorgabe für **char_kind** (für die Art des Zeichens, für dessen Klassenzugehörigkeit).

FILL_KEY_STRING ist dasjenige Wort, das der Anwender von der Tastatur aus eingibt, um Phase 2 des Programms einzuleiten. Im praktischen Betrieb, besonders wenn kurze Dateien geknackt werden sollen, wird **FILL_KEY_STRING** unvollständige Antworten liefern (einige Zeichen des Schlüssels werden stimmen, andere nicht).

Interaktives Raten - häufig nötig bei kurzen Dateien

Es gibt für den Anwender zwei Möglichkeiten, mit einem nicht stimmenden Inhalt von **KEY_STRING** umzugehen. Zum einen kann man versuchen, den Schlüsselinhalt zu erraten, zum anderen den Klartext. Häufig kann man bei näherer Betrachtung der Schlüssel-Zeichenkette englische Worte erkennen. Wenn einige Zeichen falsch erscheinen, suche man im angezeigten **KEY_CHAR** oben nach der zu dem betreffenden Zeichen gehörigen Spalte.

Diese Spalte gehe man von oben nach unten durch, um ein in Frage kommendes Zeichen zu finden. Dann verwende man **KEY_STRING!**, um **KEY_STRING** zu setzen. Schließlich lasse man sich über **SHOW_PLAIN** den resultierenden Klartext anzeigen. Man kann auch **TO_KEY_STRING** mit irgendeiner anderen **char_kind**-Routine (gefolgt von **SHOW_KEY_STRING**) verwenden, um verschiedene Filter auszuprobieren. Wir bieten eine Menge von **char_kind**-Routinen an. Man beachte, daß von den Verschlüsselnern manchmal verlangt wird, daß sie ihren Schlüssel alle Monate ändern. Häufig nehmen die Leute immer wieder denselben Schlüssel und hängen einfach die zwei Ziffern des Monats (01 für Januar usw.) an. Man halte nach solchen Mustern Ausschau.

Weiter vorn, auf Screen 25, hatten wir ein Wort namens **TRY**. Wenn wir unseren **KEY_STRING** bestimmt haben, sehen wir uns normalerweise über **SHOW_PLAIN** an, was erreicht wurde. Es kann sein, daß wir einen verständlichen Text vorfinden, bei dem aber einige Zeichen falsch sind. Diese falschen Zeichen entsprechen falschen Zeichen in **KEY_STRING**. **TRY** soll solche Falschzeichen interaktiv korrigieren.

TRY nimmt ein Klartextzeichen und einen horizontalen sowie einen vertikalen Index nach **PLAINTEXT** auf. Wir hoffen, daß dieses Klartextzeichen genau das ist, was an der betreffenden Stelle in **PLAINTEXT** stehen sollte. Die Aufspaltung in einen horizontalen und einen vertikalen Index nach **PLAINTEXT** haben wir deshalb vorgenommen, weil das **DUMP** in **SHOW_PLAIN** den **PLAINTEXT** als ein zweidimensionales Feld darstellt. Nach dem Aufruf von **SHOW_PLAIN** werden wir vermutlich **TRY** aufrufen, während wir die Ausgabe von **SHOW_PLAIN** beobachten. **TRY** korrigiert das entsprechende Zeichen in **KEY_STRING** und ruft **SHOW_PLAIN** erneut auf. Wir können ein weiteres Zeichen **TRY**en, wir können aber auch aufhören, wenn unser Klartext fehlerfrei aussieht. Das Ganze ähnelt den Wortrate-spielen, wo der Kandidat oder die Kandidatin auf ein Wort mit fehlenden Buchstaben starrt und versucht, die Buchstaben zu erraten. Sobald der Klartext fehlerfrei erscheint, führe man **SHOW_KEY_STRING** aus, um herauszubekommen, welchen Schlüssel **TRY** aufgebaut hat.

Ein Beispieldurchlauf - das Programm aus der Sicht des Anwenders

Mit der Erklärung unseres Codeknackprogramms sind wir jetzt fertig. Wenden wir uns nun einem Beispiel zu. Der Leser möge den Text in Abbildung 4 genau eingeben und ihn unter dem Namen **Message.txt** in eine Datei abspeichern. Man achte streng darauf, die End-of-Line-Zeichen an genau dieselbe Stelle zu setzen, so daß das Programm dieselben Ergebnisse liefert wie die, die wir hier beschreiben. Bei **Message.txt** müßte eine Länge von 354 Zeichen herauskommen.

Führen Sie den in Abbildung 5 gezeigten Programmteil aus, um **CIPHERTEXT** mit verschlüsselten Daten zu füllen. Jetzt nehmen Sie an, Sie wüßten nicht, wie der Klartext oder der Schlüssel aussieht, und versuchen Sie, den Code zu knacken. Führen Sie zuerst **FILL_KEY_LENGTH** aus. Sie werden ein Ergebnis bekommen, das wie in Abbildung 6 aussieht. Die Schlüssellänge ist ziemlich deutlich 13, da die Wahrscheinlichkeit hierfür 60% beträgt. Mit einer Wahrscheinlichkeit von 20% beträgt sie 26 und mit ebenfalls 20% ist es 52. Man beachte, daß beide Zahlen, 26 und 52, Vielfache von 13 sind. Werfen Sie einen Blick auf die Daten in **COINCIDENCES** oben und Sie werden bemerken, daß 13 einen Wert von 9,9% hat, wesentlich höher als die anderen Werte. Das ist ganz bestimmt eine Spitze.

Führen Sie **FILL_KEY_STRING** aus. Sie werden ein Ergebnis wie in Abbildung 7 bekommen. Das Programm hat "VeyePersonal" gefunden. Das sieht ganz gut aus, mit Ausnahme des "e" nach "Very". Sehen Sie sich die Daten von **KEY_CHAR** oben am Bildschirm an. Gehen Sie die fünfte Spalte durch. Das oberste Zeichen ist ein "e", das zweitbeste ein "-". Der Bindestrich sieht plausibel aus. Wenn Sie den Programmteil " **Very Personal**" **KEY_STRING!** **SHOW_PLAIN** ausführen, werden Sie sehen, daß das der



richtige Schlüssel ist. Statt die Spalten in **KEY_CHAR** durchzugehen, hätte man auch die *char-kind*-Filter verwenden können. Es ist ziemlich klar, daß zwischen "Very" und "Personal" ein Interpunktionszeichen oder ein Zwischenraum stehen muß. Führen Sie **' PUNCTUATION TO_KEY_STRING SHOW_KEY_STRING** aus. Damit wird **KEY_STRING** auf ein Raten über alle Interpunktionszeichen gesetzt. Sehen Sie sich das fünfte Zeichen an. Sie entdecken, daß das ein Bindestrich ist. Die zwei Methoden, mit denen man den Schlüssel erraten kann, bestehen aus dem Durchsehen der Spalten in **KEY_CHAR** und dem Ansetzen der *char_kind*-Filter.

Gehen wir zu unserem Schlüssel "VeryPersonal" zurück und versuchen wir, den Klartext zu erraten. Führen Sie **SHOW_PLAIN** aus, um sich den Klartext anzusehen. Das Ergebnis sollte wie in Abbildung 8 aussehen. Das ist ganz offensichtlich ein englischer Klartext, der ein paar falsche Zeichen enthält. In der sechsten Zeile sehen wir z.B. das Wort "pa8ers". Hier kann nur das Wort "papers" gemeint gewesen sein. Führen Sie **ASCII p 2 5 TRY** aus, um anstelle der "8" ein "p" auszuprobieren. Beachten Sie, daß wir hier den horizontalen Index 2 verwenden, da unsere Zählung bei Null beginnt. Als vertikalen Index verwenden wir 5, da die Zeilenzählung, von oben nach unten, ebenfalls bei Null beginnt. **TRY** führt, nachdem es sein **KEY_STRING**-Zeichen korrigiert hat, automatisch **SHOW_PLAIN** aus, so daß sich der Anwender das Ergebnis ansehen kann. Manchmal muß man **TRY** mehrfach anwenden, um mehrere Zeichen zu korrigieren (oder auch nur ein einziges, bei dem man sich aber nicht sicher ist, welches es denn nun sein soll).

Sobald der Klartext fehlerfrei aussieht, sehen Sie sich über **SHOW_KEY_STRING** an, welchen Schlüssel Sie sich mit den verschiedenen **TRY**-Ausführungen aufgebaut haben.

Schlußbetrachtung - die PolySub-Verschlüsselung ist ein Spielzeugalgorithmus

Probieren Sie das Programm mit verschiedenen Schlüssellängen aus. Versuchen Sie es mit "SUPERCALIFRAGILISTIC" als eine schwierige Übung und mit "UNIQUE" als eine leichte Übung. Versuchen Sie es mit einem Schlüssel, der eine Mischung aus Groß- und Kleinbuchstaben, Zahlen usw. enthält. Es macht wirklich Spaß, mit diesem Programm Codes zu knacken; so als wenn man ein Rätsel löst. Vielleicht würde es Ihnen auch gefallen, CrakPoly nach verschiedenen Richtungen hin aufzupäppeln. Es gibt Erweiterungsmöglichkeiten, die CrakPoly zum Knacken ganz kurzer Schlüssel geeigneter machen, obwohl es eigentlich schon recht gut ist. Unsere Datei Message.txt enthielt nur 354 Bytes und CrakPoly hat sie mit Leichtigkeit geknackt. Die beste Erweiterung scheint darin zu bestehen, die in **TRY** benötigten numerischen Koordinatenzeiger nach **PLAINTEXT** loszuwerden. Diese einzugeben, ist ermüdend und fehlerträchtig. Was wir bei **TRY** gern sehen würden, wäre die Möglichkeit, im Klartext über die Cursortasten mit dem Cursor herumzufahren. Wenn der Cursor dann über dem störenden

Zeichen steht, würde man das richtige Zeichen eintippen wollen und von **TRY** verlangen, daß es die Schlüsselzeichenkette in Ordnung bringt und ein neuerzeugtes **PLAINTEXT** anzeigt.

Wir hoffen, daß der Leser unsere Untersuchung über CrakPoly interessant fand. Vielleicht kann der eine oder andere das bei seinen Arbeiten sogar einsetzen. Ein Firmenchef könnte zum Beispiel ein PolySub-Programm schreiben und es seinen Angestellten mit den Worten geben: "Verwenden Sie das bei allen internen Dokumenten, bei denen die Gefahr der Werkspionage gegeben ist". Viele Angestellte werden es dann auch bei privaten Dokumenten verwenden, die sie auf den Firmencomputern gespeichert haben. Und das ist genau das, an was der Firmenchef eigentlich interessiert war. Generell gesehen, ist CrakPoly aber lediglich ein Spielzeugprogramm ohne jede kommerzielle Absicht.

CrakPoly konnte nur in Forth geschrieben werden. Es hätte niemals in C++ geschrieben werden können.

CrakPoly ist notwendigerweise interaktiv, mit **TRY** und **TO_KEY_STRING** und **KEY_STRING!** und alledem. Das ist der Grund. Ein GUI zu schreiben, das einen derartigen Grad an Interaktivität entwickelt, würde mehr Arbeit machen als für ein Programm gerechtfertigt wäre, das nur zum spielerischen Umgang gedacht ist. Alle kommerziellen Programme haben heutzutage GUI-Interfaces, und C++ ist auf das Schreiben von GUIs ausgerichtet. C++ bietet keine Möglichkeit, schnell mal einen Befehl von der Tastatur aus einzugeben. Wir hatten LEX/YACC unter C++ verwendet, um zu einem Programm mit einer Kommandozeilen-Eingabemöglichkeit zu kommen. Das ist eine wirksame Methode, sie erfordert aber einen Riesenaufwand. In Forth bekommt man das Kommandozeilen-Interface als Zugabe. Allgemein kann man sagen, daß jemand, der nur C++ kennt, zu der Überzeugung kommen muß, daß CrakPoly viel mehr Arbeit macht als es wert ist, und ein solches Projekt nie in Angriff nehmen würde. Das wäre jammerschade, denn CrakPoly hat wirklich seinen Wert.

Uns machte schon das Schreiben von CrakPoly viel Spaß und wir fanden, daß uns der Umgang mit diesem Programm ebensoviel Spaß bereitete. Außerdem ist das Schreiben und der Entwurf eines Programms, das keinen unmittelbaren Zweck erfüllen soll, auch für das Arbeiten an kommerziellen Produkten eine gute Vorbereitung. C++ mit seiner Betonung von GUIs und kommerzieller Entwicklung erfordert für ein Wochenend-Projekt einen viel zu großen Aufwand. Da es niemanden mehr gibt, der das Programmieren nur so aus Spaß an der Freude betreibt und dabei seine Programmiererfahrung erweitert, wird unsere professionelle Programmierung heutzutage mit Ausdrücken wie "Gewaltmarsch-Projekt" und "gegen jede Regel" belegt. Das scheint der Lohn der Professionalität zu sein.

Falls einer der Leser PolySub zur Verschlüsselung von irgendetwas Wertvollem verwenden wollte, sollte dieser Artikel



polyalphabetische Codes

ihn davon abhalten. Vielleicht werden wir uns nochmal an die Arbeit machen und ein Verschlüsselungsprogramm schreiben, das wirkliche Sicherheit bringt. Bis dahin empfehlen wir dem Leser, PGP zu verwenden. Das ist ein Standardprogramm, das sichere Verschlüsselungen erzeugt. Standards sind immer gut. Sie erleichtern den Austausch verschlüsselter Nachrichten von jedem zu jedem. Sich auf PolySub als Standard zu einigen, nur weil es allgemein bekannt ist, wäre jedoch ein Fehler.

Abbildung 4
AMENDMENT 4.

The right of the people to be secure in their persons, houses, papers, and effects, against unreasonable searches and seizures, shall not be violated, and no warrants shall issue but upon probable cause, supported by oath or affirmation, and particularly describing the place to be searched, and the persons or things to be seized.

Abbildung 5

```
" MESSAGE.TXT" PLAINTEXT INPUT_FILE
" Very-Personal" KEY_STRING! ENCRYPT
```

Abbildung 6

1) 2.5	2) 3.4	3) 0.8	4) 1.7	5) 1.7	6) 0.8	7) 0.2	8) 1.7
9) 1.7	10) 1.1	11) 1.1	12) 1.4	13) 9.9	14) 2.3	15) 2.0	16) 1.7
17) 1.4	18) 2.6	19) 2.6	20) 1.1	21) 1.8	22) 0.6	23) 1.5	24) 1.2
25) 1.2	26) 4.5	27) 1.2	28) 2.1	29) 1.2	30) 0.3	31) 2.1	32) 0.9
33) 0.0	34) 3.4	35) 1.2	36) 1.2	37) 0.9	38) 1.5	39) 6.3	40) 0.9
41) 0.3	42) 1.2	43) 0.3	44) 2.9	45) 0.6	46) 2.2	47) 2.9	48) 0.9

Schwellwert = 4.9

1) 0.0	2) 0.0	3) 0.0	4) 0.0	5) 0.0	6) 0.0	7) 0.0	8) 0.0
9) 0.0	10) 0.0	11) 0.0	12) 0.0	13) 60.0	14) 0.0	15) 0.0	16) 0.0
17) 0.0	18) 0.0	19) 0.0	20) 0.0	21) 0.0	22) 0.0	23) 0.0	24) 0.0
25) 0.0	26) 20.0	27) 0.0	28) 0.0	29) 0.0	30) 0.0	31) 0.0	32) 0.0
33) 0.0	34) 0.0	35) 0.0	36) 0.0	37) 0.0	38) 0.0	39) 0.0	40) 0.0
41) 0.0	42) 0.0	43) 0.0	44) 0.0	45) 0.0	46) 0.0	47) 0.0	48) 0.0
49) 0.0	50) 0.0	51) 0.0	52) 20.0	53) 0.0	54) 0.0	55) 0.0	56) 0.0
57) 0.0	58) 0.0	59) 0.0	60) 0.0	61) 0.0	62) 0.0	63) 0.0	64) 0.0
65) 0.0	66) 0.0	67) 0.0	68) 0.0	69) 0.0	70) 0.0	71) 0.0	72) 0.0
73) 0.0	74) 0.0	75) 0.0	76) 0.0	77) 0.0	78) 0.0	79) 0.0	80) 0.0
81) 0.0	82) 0.0	83) 0.0	84) 0.0	85) 0.0	86) 0.0	87) 0.0	88) 0.0
89) 0.0	90) 0.0	91) 0.0	92) 0.0	93) 0.0	94) 0.0	95) 0.0	96) 0.0
97) 0.0	98) 0.0	99) 0.0					

Zu groß = 0.0

Länge des Schlüssels ist: 13

Historische Highlights der Kryptografie

Zeit	Ort	Bemerkung
1900 v. Chr.	Ägypten	Verwendung von Nicht-Standard-Hieroglyphen.
500-600 v. Chr.	Palästina	ATBASH (Zeichen-Substitution mit umgekehrtem Alphabet).
725-790 n. Chr.	Persien	Abu `Abd al-Rahman al-Khalil ibn Ahmad ibn `Amr ibn Tammam al Farahidi al-Zadi al Yahmadi beschreibt in einem Buch stolz die geglückte Entschlüsselung eines für den byzantinischen Kaiser bestimmten griechischen Code.
1250		Roger Bacon schreibt "A man is crazy who writes a secret in any other way than one which will conceal it from the vulgar".

Abbildung 3

Elephants are very eloquent.
Especially Penelope!

Abbildung 7



```
V e r y e e 6 o $ ?
! 8 - ! ! & / 3 l
+ < * o ) 1 ' ; & )
* = - } $ 3 0 * & "
, " < y , 7 : , * . (
l 6 , ~ 7 ; s . + 2
H X / ! * = ! = 5 *
Z # 4 b = + > # # > a -
7 $ 0 7 c - " & ' ! % .
' 3 ; d 1 % l ( ' - <
- 4 = h 6 & 2 5 4 / >
6 7 ? i ' ; : : m `
> l } O : < = < o A
i x P ! = ? n ! !
{ O _ S " r c z " #
| ! # I ! # # ^ B D # $
X " $ ! " % $ E " ' %
% % " " # & ( " " # ( &
```

1397	Italien	Auf Wunsch von Clemens VII. erfindet Gabrieli di Lavinde den Nomenklatur-Code, der wegen seiner Einfachheit die nächsten 450 Jahre vor allem in diplomatischen Kreisen verwendet wird.
1412	Arabien	Die Subh al-a`sha, eine 14-bändige arabische Enzyklopädie beschreibt auch kryptografische Methoden (Substitution, Transposition, erstmals mehrere Substitutionen für ein plaintext-Zeichen!).
1466	Italien	Leon Battista Alberti erfindet erste Chiffrierscheibe.
1518	Deutschland	Erstes gedrucktes Buch über Kryptografie von Johannes Trithemius; bei einem der Algorithmen sieht der Cipher text auf den ersten Blick wie ein Gebet aus.
1553	Italien	Verwendung einer „passphrase“ als Schlüssel (meist falsch als Vigenère bezeichnet). Beispiel aus dem Beginners' Guide for Cryptography.
1558	Frankreich	Blaise de Vigenère: erste „autokey“-Systeme (entspricht CBC und CFB).

```
0 1 2 3 4 5 6 7 8 9 A B C D E F 0123456789ABCDEF
0017:015C10 56 65 72 79 65 50 65 72 73 6F 6E 61 6C VeryePersonal
```

Abbildung 8

```
0 1 2 3 4 5 6 7 8 9 A B C D E F 0123456789ABCDEF
0017:096390 41 4D 45 4E 0C 4D 45 4E 54 20 34 2E 0D 0A 0D 0A AMEN.MENT 4.....
0017:0963A0 54 20 65 20 72 69 67 68 74 20 6F 66 20 74 20 65 T e right of t e
0017:0963B0 20 70 65 6F 70 6C 65 20 74 6F 20 2A 65 20 73 65 people to *e se
0017:0963C0 63 75 72 65 20 69 6E 20 3C 68 65 69 72 20 70 65 cure in <heir pe
0017:0963D0 72 73 6F 6E 73 64 0D 0A 68 6F 75 73 65 73 2C 20 rsonsd..houses,
0017:0963E0 70 61 38 65 72 73 2C 20 61 6E 64 20 65 66 66 2D pa8ers, and eff-
0017:0963F0 63 74 73 2C 20 61 67 61 69 6E 73 74 68 75 6E 72 cts, againsthunr
0017:096400 65 61 73 6F 6E 61 62 6C 65 68 73 65 61 72 63 68 easonablehsearch
0017:096410 65 73 0D 0A 61 6E 2C 20 73 65 69 7A 75 72 65 73 es..an, seizures
0017:096420 2C 20 73 20 61 6C 6C 20 6E 6F 74 20 62 65 20 76 , s all not be v
0017:096430 21 6F 6C 61 74 65 64 2C 20 61 6E 64 20 26 6F 20 !olated, and &o
0017:096440 77 61 72 72 65 6E 74 73 20 73 20 61 6C 6C 0D 0A warrents s all..
0017:096450 69 73 73 75 65 20 62 3D 74 20 75 70 6F 6E 20 70 issue b=t upon p
0017:096460 72 6F 62 61 2A 6C 65 20 63 61 75 73 65 2C 20 73 roba*le cause, s
0017:096470 75 38 70 6F 72 74 65 64 20 62 79 20 6F 61 3C 68 u8ported by oa<h
0017:096480 20 6F 72 0D 0A 61 66 66 69 72 6D 29 74 69 6F 6E or..affirm)tion
0017:096490 2C 20 61 6E 64 20 70 61 3A 74 69 63 75 6C 61 72 , and pa:ticular
0017:0964A0 6C 79 20 64 65 3B 63 72 69 62 69 6E 67 20 74 68 ly de;cribing th
0017:0964B0 65 20 38 6C 61 63 65 20 74 6F 20 62 65 0D 0A 3B e 8lace to be..;
0017:0964C0 65 61 72 63 68 65 64 2C 20 61 6E 64 68 74 68 65 earched, andhthe
```

1790	USA	Thomas Jefferson erfindet „wheel cipher“, 1922 wiederentdeckt und von der US-Marine bis in den 2. Weltkrieg verwendet
1854	England	Charles Babbage erfindet „wheel cipher“ erneut
1891	Frankreich	Major Etienne Bazeries verwendete mehrere „wheels“: eine der ersten Chiffriermaschinen (Bazeries-Zylinder)
1917	USA	Gilbert S. Vernam entdeckt „one-time-pad“, der später noch bei der Entschlüsselung der Enigma eine Rolle spielen sollte.
1923	Deutschland	Arthur Scherbius gründet „Chiffriermaschinen AG“ und vermarktet Enigma
1930-50	weltweit	Jedes größere Land erfindet eigene Chiffriermaschinen: England: TYPEX Japan: PURPLE (aus Telefon-Relais) USA: SIGABA (M-134-C, ECM Mark II)



Farbe ins Leben

Martin Bitter
Mehrhoog

Farbig zu drucken kann sich unter Win32For problematisch gestalten. Aber wenn man einen Farbdrucker hat, dann will man den auch nutzen, auch dann, wenn man sich dazu mit Windows UND Win32For tiefer auseinandersetzen muß

Stichworte: Win32For, Printer, Opcode,

Neben der Möglichkeit, seine eigenen Druckbefehle zu schreiben, bietet Win32For 'natürlich' schon eine ganze Reihe 'eingebauter' Druckbefehle.

Es gibt ein vordefiniertes Objekt ThePrinter, das mit der Klasse WinPrinter erzeugt wurde.

WinPrinter verfügt über viele Methoden, die sich noch einmal in Methodengruppen unterteilen lassen und die unterschiedliche Philosophien im Umgang mit Druckern widerspiegeln (Wow: widerspiegeln und Philosophie in einem Satz, der Alt68'er ;-).

Es gibt die Möglichkeit, Befehle direkt zum Drucker zu senden: direkte Druckerbefehle.

Es gibt die Möglichkeit, mittels entsprechender Zeichen- und Textbefehle in eine Bitmap (Speicherbereich) zu zeichnen und dann, wenn alles fertig ist, diese Bitmap auszudrucken. Zusätzlich gibt es noch die Möglichkeit, alle Druckbefehle 'aufzuheben' und erst wenn es gewünscht wird, diese Befehle abzarbeiten und zum Drucker(treiber) zu senden.

KleinWeich übernahm dazu das Konzept der Metafiles: Gedacht ist es so: Egal, welches Ausgabegerät tatsächlich angesprochen wird, die grundlegenden Befehle werden gleichartig definiert. Eine Linie wird mit dem Befehl (der Methode) Line: gezeichnet. Dieser Methode müssen (mindestens) vier Parameter übergeben werden (die Koordinaten des Start und des Endpunktes). Gibt man nun als Ziel nicht einen physikalischen oder virtuellen Drucker an, sondern schreibt den Befehl selbst in ein Metafile, so kann dieses Metafile von einem x-beliebigen Rechner verarbeitet und auf den ausgewählten physikalischen Geräten ausgegeben werden.

So weit die Theorie: erfolgreich umgesetzt wurde der Umgang mit Metafiles auf einigen Betriebssystemen, aber (leider) wenig erfolgreich auf Win95/98 Systemen. Wer schon einmal versucht hat, die Windows EnhMetaFile Funktionen zum Datenaustausch zwischen verschiedenen Rechnern zu benutzen weiß das!

Ob Tom Zimmer daran dachte, als er sein Objekt ThePrinter schrieb, weiß ich nicht. Jedenfalls hat er ein sehr ähnliches Konzept angewendet, das nun als dritte Möglichkeit zum Umgang mit dem Drucker zur Verfügung steht!

Alle Methoden, die mit dem Präfix Printer... oder LPT...

beginnen sind 'Meta' Befehle, die nicht ausgeführt, sondern in eine Liste gespeichert werden. (z.B. PRINTERSETPIXEL, PRINTERLINECOLOR, PRINTERTEXTOUTFONT, LPTPRINTERTEXTOUT).

Darüber habe ich mir nie weiter Gedanken gemacht, bis ich farbige Texte drucken wollte.

Zwar erschienen die Texte in gewünschter Größe an der richtigen Stelle, aber alle Farbbefehle wurden ignoriert.

So hatte ich es versucht: Blue SetTextColor: ThePrinter. Diese Sequenz wurde auch anstandslos (keine Fehlermeldungen etc.) geschluckt, oder besser formuliert verschluckt - denn die Textfarben änderten sich nicht.

Letztendlich stellte sich heraus, SetTextColor: ist eine direkte Methode, die sofort ausgeführt wird, und das erklärt schon alles.

Beim Aufruf BLUE SetTextColor: ThePrinter wird die Textfarbe des Druckers sofort auf blau gesetzt, der folgende Befehl 100 100 s" hallo!" bold_font PRINTERTEXTOUTFONT: ThePrinter wird 'nur' gespeichert. Der nächste Farbwechsel, z.B. LTGREEN SetTextColor: ThePrinter, wird ebenfalls sofort ausgeführt und nicht in der Liste der Druckerbefehle gespeichert. Bei der Druckerinitialisierung vor dem endgültigen Ausdruck, wird die Textfarbe auf schwarz gesetzt und alle Farbeinstellungen und Farbwechsel sind ins Nirwana gegangen.

Lösung: eine neue Methode muß her.

Tom Zimmer hat insgesamt 18 solcher 'Metabefehle' definiert, die einzeln als Opcode durchnummeriert sind.

Beim Aufruf der Methode PRINTERLINETO: geschieht folgendes:

In die Liste 'drawlist' wird ein Count eingetragen, hier ist es 5, dann wird die Nummer des entsprechenden Befehls, also der Opcode 1, hier definiert als Konstante P_LINETO, eingetragen, und im Anschluss daran werden die Word-Parameter vom Stack geholt und ebenfalls in die Liste eingetragen. (Rechnung: 1 Byte Befehlsnummer + zwei Word-Parameter = 5 Byte = Count)

Der irgendwann erfolgende, abschließende Druckbefehl arbeitet via count die Liste 'drawlist' dieser Druckbefehle ab und stößt dabei auch auf den Wert 1 (Opcode), was ihn dazu veranlasst, die folgenden 4 Byte als je zwei 16-bit Parameter auf den Stack zu legen und dann den eigentlichen Druckbefehl p-lineto aufzurufen.

Ausgestattet mit diesem Wissen kann ich nun beliebige weitere sinnige und unsinnige Druckerbefehle einbauen.

Eins bleibt noch zu überlegen: Soll ich eine neue Klasse Farb_WinPrinter und dann ein neues Objekt TheColPrinter erzeugen? Das ließe die ursprüngliche Klasse WinPrinter und das Objekt ThePrinter unberührt und soll ja einer der Vorteile von OOP sein - nicht eine Klasse ändern, sondern Vererbungsmechanismen und Überdefinieren von Methoden nutzen.

Ginge es nur darum, eine neue Methode PrinterSetTextColor: zu erzeugen, wäre das ja noch einfach, aber die Befehle zum Erzeugen der 'drawlist' und ihrer Interpretation sind interne Befehle der Klasse WinPrinter und daher gekapselt! Jede



Menge stupider Tipp- oder paste-and-copy Arbeit müsste geleistet werden - schönes OOP!

Weniger Arbeit, weniger Speicherplatz wird verbraucht, wenn ich direkt in den Quelltext gehe, in der Datei DC.F einige Zeilen einfüge, und das System neu metacompiliere - so hab ich es letztendlich getan.

(Dies in die Reihe der Opcodes eingeben. Einfach nach 17 constant P_STROKE-ANDFILL suchen)

```
18 constant P_SETTECTCOLOR \ ein
neuer Opcode M.B. 1999
```

(Dies bei den Printermethoden einfügen: Suche nach: :M PrinterLineColor:)

```
:M PrinterSetTextcolor
  { color_object -- }
  \ eine neue Methode
  \ M.B. 1999
  color_object ?ColorCheck
  drop
  drawing?
  if 5 d-c, \ record length
    \ is 5 bytes
    \ + count
  P_SETTEXTCOLOR to lastcall
  P_SETTEXTCOLOR d-c,
    \ opcode
  color_object d-,
    \ compile the
    \ color obj
  then ;M
```

(Dies hinter p-linecolor eingeben)

```
\ p-SetTextcolor by M.B.
: p-settextcolor ( a1 n1 -- a1 n1 )
  print-page? 0= ?exit
  ?page-started
  over 1+ @ ( color_ref )
  SetTextColor: self ;
```

\ in die Methode PrinterEnd: muss an der Stelle wo auf die Opcodes abgefragt wird folgende Zeile eingesetzt werden:

```
P_SETTEXTCOLOR of p-settextcolor endof
\ colored text M.B. 1999
```

Eine interessante URL:

www.setiathome.ssl.berkeley.edu



Forth Interest Group International (FIG USA)

Wollen Sie mit der ganzen Welt verbunden sein und dabei Ihr Englisch perfektionieren ? Amerika ist ein wesentlicher Teil der ganzen Welt. Zumindest, was Forth betrifft. Über tausend Mitglieder aus allen Ländern sind bei uns.

Werden Sie auch ein Mitglied in der Amerikanischen Forth-Gesellschaft (FIG-USA).

Für 45 Dollar im Jahr (Studenten zahlen 18 Dollar) bekommen Sie 6 Hefte unserer Vereinszeitschrift Forth Dimensions und genießen auch sonst verschiedene Vorteile. In den Heften erfahren Sie Forth-Neuigkeiten aus aller Welt, neue Produkte, Literatur, Forth-Ideen, fundiertes Wissen, Artikel auch für Einsteiger, Projekte, Leser-Diskussionen, Quelltexte, Hinweise auf Internet-Verbindungen, kostenlose Forth-Systeme und vieles mehr. (Für Übersee-Porto müssen wir leider noch 15 Dollar hinzurechnen).

Unmittelbare Informationen über uns bekommen Sie, wenn Sie auf der Homepage der Deutschen Forth-Gesellschaft "Links zu anderen Forth-Organisationen" und dann "Forth Interest Group (USA)" anklicken.

Ansonsten bekommen Sie Auskünfte über das amerikanische Forth-Büro:

Forth Interest Group
100 Dolores Street, suite 183
Carmel, California 93923
USA

oder auch vom Redakteur, Marlin Ouverson, unter der E-mail Adresse:

E-Mail: office@forth.org oder
editor@forth.org

Gehaltvolles

zusammengestellt und übertragen
 von Fred Behringer

Forth Dimensions der Forth Interest Group, USA

Januar/April 1999

Diesmal ein Doppelheft (80 Seiten). Marlin Ouverson bittet um Entschuldigung dafür, daß die Verzögerung noch größer geworden ist. Nicht zuletzt auch darum, weil die Autoren ihre Beiträge nur zögernd einreichen. Er ermutigt ausdrücklich auch die "ausländischen" Forth-Freunde, Beiträge einzuschicken. Ich finde die Doppelheft-Idee gut. Damit gelingt es sicher, aufzuholen und den Anschluß wieder zu finden. Und das nächste Doppelheft ist schon in Bearbeitung - sagt Marlin.



Rezension der Forth Dimension

Dem Heft liegt eine Einladung zur FORML-Konferenz (19.-21. November 1999) bei. Die ist natürlich bei Erscheinen der vorliegenden VD längst überholt. Aber das Wissen um die folgenden drei Punkte ist dann sicher immer noch interessant: (1) Es geht um "Forth und das Internet", (2) Beiträge von ausserhalb der USA werden ganz besonders begrüßt und gern gesehen, (3) 575 Dollar im Doppelzimmer sind für unsereins, ganz abgesehen vom eigentlichen Flug, den man vielleicht mit einem 2-Wochen-Urlaub verknüpfen könnte, doch ein bißchen happig.

Das Heft enthält 4 interessante Artikel, eigentlich Programmpakete, von Wil Baden. Er setzt darin seine bekannten Serien "Toolbelt" (Forth-Werkzeugtasche) und "Stretching Forth" (Forth-Ausbau) fort.

Im Editorial Abschiedsworte für einen hervorragenden Forthler, früheren Präsidenten der FIG US und Direktor der jährlichen FORML-Konferenzen: Robert Reiling. "Er glaubte, seine schwere Krankheit besiegt zu haben, und organisierte mit Zuversicht die kürzliche 20. FORML-Konferenz. Die Krankheit war stärker."

5 High Accuracy Table Lookup Using Cubic Interpolation

Brad Eckert <bradbits@hotmail.com>

Kubische Interpolation (Polynome 3. Grades) zur Approximation beliebiger Funktionen in einem vorgegebenen Intervall. Verwendet nur ganze Zahlen und setzt einige Vernachlässigungen ein, um die Mathematik einfach zu halten. Text nicht viel mehr als das Vorliegende, 1½ Seiten Programm. High-Level-Forth. Im Beispiel liefern 19 Interpolationenpunkte für den Hauptteil der Sinusfunktion eine Genauigkeit von mehr als 16 Bit.

7 User Stacks in ANS Forth

Leonard Zettel <zettel@acm.org>

Der Autor präsentiert Worte zur Einrichtung eines Benutzer-Stacks als verkettete Liste: Stacks als abstrakte Datentypen. 2 Seiten Text, 8 Colon-Definitionen.

9 Forth-Gesellschaft eV

Eine gut hervorgehobene Meldung über unsere veränderte Web-Adresse. In übergroßen Buchstaben und mit Umrahmung. Danke, Marlin!

10 Aspects of a Particular Three-Stack Machine Design

Rick Hohensee <rickh@capaccess.org>

Wieviele Stacks braucht Forth - kann es überhaupt gebrauchen? Zwei sind zu wenig, 1024 sind zu viel. Der Autor versucht es mit drei: Return-Stack, Daten-Stack und Pointer-Stack. Wesentlich ist das SIZE-Register, das die gerade einge-

schaltete Datentypbreite für den Daten-Stack festhält: "Der Daten-Stack arbeitet mit Pytes, Gruppen von 1, 2, 4, 8, 16 ... 256 Bytes. Boolesche Flags werden durch das unterste Byte im Pyte wiedergegeben. FALSE ist Null, Nichtnull ist TRUE."

2+ , 3DUP und dergleichen gehören beim Autor der Vergangenheit an. 8DUP , 64DUP , 128XOR , 11+ sind auf seiner virtuellen 3-Stack-Maschine schon ohne zusätzliche Namensakrobatik vorhanden und leicht verwendbar.

Der Autor "kommt von der Landvermessung her, wo man Zahlen nur paarweise betrachtet, Meßergebnis und Genauigkeit". (Meinung des Rezensenten: Jeder gewissenhafte Physiker denkt so, jeder Maschinenbauer oder Architekt sollte so denken.) Pyte = Precision Byte. SIZE steht für Genauigkeit: 0 = "über den Daumen gepeilt" (real-malende Künstler), 2 = Rechenschieber. Alles in einer Maschine, allein durch den Wert von SIZE geregelt und regelbar.

"Logik und Arithmetik dieser (virtuellen) Maschine arbeiten mit beliebigen SIZE-Werten von 1 bis 256." Der Rezensent: Erstaunlich!

Ganz dumme Frage des Rezensenten: Was haben eigentlich die drei Stacks mit der veränderlichen Typenbreite zu tun? Einführung von SIZE für den Daten-Stack würde doch reichen? Ich merke mir das für die eigenen laufenden Vorhaben vor.

Herzerfrischend der lockere Stil der Sprache (der Rezensent und Übersetzer hat noch viel zu lernen): "I kinda like the term atoms in lieu of the usual primitives." "In lieu" in place of "in place" ist nicht kritisch, vornehm geht die Welt zugrunde, aber suchen Sie mal "kinda" im Wörterbuch! In der eigenen Sprache setzt man sowas gern ein - und kommt sich richtig wichtig dabei vor.

Sehr beeindruckend andererseits aber auch die Präzision, mit der die Ideen in diesem Artikel mit wenigen gezielten Worten "rübergebracht" werden (Flapsigkeit allein reicht eben nicht). Nur ein Beispiel: "It is similar to what has been called 'call threading' (by Ertl or Paysan, I think, in comp.lang.forth)."

Es ist kein Zufall, daß ich diesem Artikel wesentlich mehr Worte gewidmet habe als beiden vorhergehenden zusammen. Autor ist halt nicht gleich Autor. Aber bei Forth bekommt jeder seine Chance.

15 Polyalphabetic Encryption Cracker

Hugh Aguilar <vaguilar@dancris.com>

Ein Know-How-Artikel. 4 Seiten Text, 7 Seiten Programm. Für die Besprechung ist sicher alles gesagt, wenn ich den ersten Abschnitt vollständig übersetze:

PolySub - bekannt, aber nicht sehr sicher. In diesem Arti-



kel erklären wir die Polyalphabetic Substitution Cipher (der Rezensent: ich laß das mal so stehen; es wird gleich klar, was gemeint ist), die wohl die meisten Leser kennen werden (der Rezensent: hört, hört!). Das ist jenes Chiffrierverfahren, bei dem ein bestimmtes Schlüsselwort wiederholt mit dem Klartext geXORt wird, um den Chiffretext zu erzeugen (oder umgekehrt). Wir stellen dann ein Programm vor (der Rezensent: selbstverständlich in Forth, High-Level-Forth), das diesen Code knackt, unter der Voraussetzung allerdings, daß die Buchstaben und Zeichen im Klartext mit unterschiedlicher Häufigkeit auftreten und daß insbesondere ein bestimmtes Zeichen wesentlich häufiger erscheint als alle anderen (wir nehmen an, daß das der Zwischenraum ist). Der vorliegende Artikel wendet sich an den Neuling. Wir erklären daher unser Codeknackprogramm auf der Implementationsebene anhand von vielen Programmbeispielen. Auf dem Computer wird PolySub normalerweise über XOR implementiert. Damit kann ein und dasselbe Programm sowohl zum Verschlüsseln als auch zum Entschlüsseln verwendet werden (man beachte, daß XOR sich selbst rückgängig macht: $a \text{ XOR } b \text{ XOR } b \text{ XOR } a = a$). In der Zeit vor Erfindung des Computers hat man Plus und Minus verwendet. PolySub wurde im amerikanischen Bürgerkrieg von der Unionsarmee ausgiebig verwendet.

26 PRESWOOP

Wil Baden <wilbaden@netcom.com>

"Rick VanNorman nahm mein 'Simple Object-oriented Programming' und baute es aus. Jetzt ist es wesentlich leistungsfähiger. Wegen der verschiedentlichen Extras ist es nun zwar eine nicht mehr ganz so 'einfache' (simple) Implementation, aber es ist immer noch leicht zu handhaben und schnell." Der Autor zeigt mit 2½ Seiten Programm, wie man Rick VanNormans Erweiterung (für SwiftForth - siehe Seite 29) ANSifizieren kann. "MARKER macht bei 'Word-List-Identifiers' Schwierigkeiten." Der Autor bessert das aus.

29 SWOOP: Object-Oriented Programming in Swift-Forth

Rick VanNorman <rvn@forth.com>

Rick VanNorman ist mir von seinem OS/2-ANS-Forth her als Autor bekannt. Das gibt es schon seit einigen Jahren. Es ist über einen mitgelieferten DOS-Extender (alles kostenlos) auch von der DOS-Ebene aus betreibbar. Friederich Prinz hat mich seinerzeit darauf aufmerksam gemacht. Warum hört man nicht mehr davon? Weil die c't nichts mehr über OS/2 schreibt und es dort und anderswo "Gurus" gibt, die meinen, OS/2 sei "out"? Als ob ein Betriebssystem, mit dem ich laufend experimentiere und dessen Ideen ich ausschachte, jemals out sein könnte! Ideen sterben nicht, Betriebssysteme sind für den, der lernen möchte, niemals out.

Rick ist, wie im Vorspann gesagt wird, 20 Jahre bei Forth, ein "Forth-Zigeuner". Kürzlich ist er der Firma Forth-Inc. (Elizabeth Rather) beigetreten. Hier also der Artikel, auf den Wil Baden auf Seite 26 in Form einer ANSifizierung auf-

merksam macht. Eine Warnung an die ewig Eiligen voraus: 6 Seiten (ausführlicher und klarer) Text, 11 Seiten Programm. Das will im allgemeinen noch nicht viel heißen. Hier heißt es aber Tiefe und Vollständigkeit.

Bevor der Autor dieses Projekt in Angriff nahm, sah er sich mehr als 4 OOP-Implementationen genauer an. Keine fand seine uneingeschränkte Zustimmung. (Er untersuchte 5 Veröffentlichungen aus der Forth Dimensions und 2 englischsprachige Bücher. - Im vorliegenden FD-Doppelheft sind 13 Artikel enthalten, von denen 10 keine Literatur angeben. 5-mal wird die Forth Dimensions zitiert, 10-mal andere englischsprachige Literatur und einmal (1-mal) eine technische Schrift aus Deutschland, ganz offensichtlich ebenfalls englisch verfaßt.)

Was der Autor in den von ihm untersuchten Systemen (Baden, Ertl, Pountain) nicht fand und hier behandelt: "Der erzeugte Code sollte zielkompatibel sein." Weitere Punkte aus dem Inhalt: Dynamische Objekte, eingebettete Objekte, Informationsverbergung, Vererbung und Polymorphismus, Datenstrukturen, Klassen, Elemente, Implementationsstrategien, globale Zustandsinformation, 'THIS enthält die Kennung (handle) der aktiven Klasse, 'SELF enthält die Datenadresse des aktiven Objekts, Compilationsstrategien, Compilation von Klassen und Objekten, SELF als Objekt-Statthalter, frühe Bindung, späte Bindung, Optimierung, SwiftForth 2.0 ist momentan in der Beta-Phase und wird SWOOP enthalten.

46 Embedding 4tH Bytecode

Hans Bezemer <hansoft@bigfoot.com>

"Bytecode" als andere Bezeichnung für Zwischencode. Ganz allgemein. Wie bei UCSD-Pascal oder neuerdings Java. "Bytecode ist Code, der von einem virtuellen Prozessor (meist 'virtuelle Maschine' genannt) ausgeführt wird." Der Autor pflegt sein 4tH seit geraumer Zeit. Der 4tH-Compiler kann (auch) Bytecode erzeugen, den man in ein C-Programm einbetten kann (Thema der letzten FORML-Tagung). Der Bytecode wird von einer virtuellen Maschine in Form eines kleinen C-Programms interpretiert. "Das dauert zwar etwas länger, erhöht aber die Portierbarkeit sehr." 1½ Seiten Text, 2½ Seiten Programm. 4tH downloadbar von ftp.taygeta oder von <http://visitweb.com/hansoft>, der Webseite des Autors.

50 PIC Assembler

Richard M. Mayer <rmayer123@aol.com>

Ein Einphasenassembler für den Mikrocontroller PIC17C44, der auf F-PC 3.60 arbeitet. Keine Sprungweitenbeschränkung (ein Thema, das dem Rezensenten nicht ganz fremd ist). Verwendet Nummern statt Namen als Indizes in eine Tabelle mit GOTO-Adressen. Hier nur Kostproben. Das vollständige Public-Domain-Programm kann vom Autor bezogen werden. Der Autor, Elektroingenieur, beschäftigt sich seit 1978 mit Forth und hat von der ersten Stunde des Erscheinens von F-PC an dieses verwendet. Seit 1994 ist er in Pension.



Rezension der Forth Dimension

52 Linked List and Ordered List Wil Baden <wilbaden@netcom.com>

"Eine verkettete Liste ist eine Folge von Adressen, in der jede Adresse die nächste Adresse enthält, bis auf die letzte, die 0 enthält." Ein ganzes Linked-List-Paket. 5½ Seiten Programm, alles Colon-Definitionen. Objektorientiert, mit Zuschnitt auf SwiftForth. Verwendet SWOOP (siehe Seite 26 und 29). Die "geordnete" Liste im Sinne von mehrfach geordneten Listen.

58 Ordered List Examples Wil Baden <wilbaden@netcom.com>

Eine Textdatei wird eingelesen und ihre Zeilen werden sortiert angezeigt. Sortieren erfolgt durch Einordnen in eine geordnete Liste. Eine wesentliche Rolle spielen dabei Worthäufigkeiten. 5 Seiten Programme.

65 Reed-Solomon Error Correction (Part Two) Glenn Dixon <Dixong@iomega.com>

Zweiter Teil des äußerst interessanten Artikels über die Reed-Solomon-Fehlerkorrektur. Der erste Teil erschien in Übersetzung auch bei uns (VD 3/99). Und auch dieser Teil erscheint wieder in Übersetzung (siehe andernorts in der vorliegenden VD). Im ersten Teil wurde die Mathematik der endlichen Körper (Galoisfelder) besprochen. Hier nun geht es um die (vorausschauende) Fehlerkorrektur selbst. Das vollständige Programm, in dem ein Beispiel behandelt wird, kann vom Autor auf elektronischem Wege bezogen werden. Noch ganz schnell der letzte Abschnitt des Artikels in Übersetzung:

"Zu sagen bleibt noch, daß mir (*dem Autor*) die interaktiven Möglichkeiten in Forth eine wertvolle Hilfe beim Verstehen dieses schwierigen Themas waren. Einfach hingehen und Arithmetik in endlichen Körpern von der Tastatur aus zu betreiben, so als ob es ein spezieller Rechner wäre, und mit den Algorithmen herumzuspielen, war schon eine schöne Sache."

Der Rezensent darf hinzufügen: Jetzt, da ich einen der Entwickler nicht nur lesen, sondern sogar übersetzen durfte, ist mein Vertrauen in die Firma IOMEGA ins Unendliche gestiegen. Ich habe mir sofort zwei weitere ZIP-100M-Laufwerke gekauft. Diesmal in ATAPI-Ausführung, für den zweiten Controller-Strang, parallel zum CD-Laufwerk. Geht prima (unter DOS, Win3.11 und Win95) und kostet in München nur noch DM 138,- - in "Bulk"-Ausführung, was immer die meist nur unzulänglich Deutsch sprechenden ComputerVerkäufer unter diesem Kauderwelsch persönlich verstehen mögen.

68 Look Ma, no interrupts! Real-Time Forth Everett F. Carter Jr. <skip@taygeta.com>

Ein Artikel des Präsidenten der FIG US, den dieser schon 1998 auf der FORML-Konferenz vorgetragen hatte.

Echtzeit-Forth wurde früher meist in Silizium gegossen, Forth und Betriebssystem in einem. Zunehmend geht man nun aber auch dazu über, Echtzeit-Forth zu betreiben, ohne gleich ein ganzes Betriebssystem mit dazu hochziehen zu müssen. "Skip" Carter zeigt, wie man's macht, und liefert ein kleines Beispiel unter Echtzeit-Linux.

Stichworte: RTKernel und VxWorks als speziell zugeschnittene Betriebssysteme, Kernel-Patch und Zugang zur Echtzeit-API-Bibliothek unter Linux oder Windows NT als Allzweckbetriebssysteme. Als Beispiel bringt der Autor eine Anwendung, bei der zwei Servomotoren mit Impulsbreitenmodulation am Parallelport eines PCs auftreten. Der Echtzeitteil ist in C gehalten und kommuniziert mit Forth über einen Unix-Dateideskriptor (flexibler als das File-Wordset aus ANS-Forth). - 1 Seite Text, 1 Seite Glossar, 4 Seiten Programm.

75 EVALUATE Macros Wil Baden <wilbaden@netcom.com>

(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



13 Seiten Programm, in ANS-Forth mit Zuwendung zu Swift-Forth. Ein Überblick über die bisherigen Ideen des Autors zum Thema "Makros". Er erklärt, weshalb er EVALUATE dem POSTPONE vorzieht. "EVALUATE stammt (in der Form von EVAL) von Martin Tracy und wurde der Forth-Gemeinde durch John Hayes nähergebracht." Der Autor hatte und hat eigene Ideen.

78 Forth in Space - again

Im Röntgenstrahl-Raumteleskop Chandra (gestartet am 23.7.99) wurde Forth eingesetzt. Mehrere Mitglieder der FIG US haben mitgewirkt, darunter Tom Zimmer (ohne dessen Wissen) mit seinem TCOM.

FORTHWRITE der FIG UK, Großbritannien

Nr. 104 November 1999

Das ist das letzte Heft im vorletzten (!) Jahr dieses Jahrtausends. An der Zusammensetzung des Führungsstabs in der FIG UK scheint sich seit der letzten Jahresversammlung nichts geändert zu haben. Die Mitgliederbeiträge sind geringfügig gestiegen. Aber wirklich nicht der Rede wert. Erste "Preiserhöhung" seit 14 Jahren! Nimmt sich gegenüber "Germany and USA", wie Chris Jakeman sagt, "immer noch recht vorteilhaft aus". Mehr davon demnächst. - Die IRC-Stitzungen (Internet Relay Chat) beginnen ab sofort schon um 22.00 Uhr unserer deutschen Zeitrechnung (O-Ton Chris: "to give Fred and Martin better access times"). Jeden ersten Samstag im Monat im IRC-Net-Kanal #FIGUK. (Wenn schon nicht bei uns, warum eigentlich nicht bei den Freunden in England? Ist doch ein prima Gefühl! So ganz "international"! Und das bißchen Englisch können wir doch alle. Laßt Euch doch nicht lumpen! Und wenn Ihr wüßtet, wieviele Fehler DIE machen. In der Hast und in der ständigen Angst, bei dem unmenschlichen Tastaturgehacke hinten an zu bleiben. Da fallen unsere paar Fehler überhaupt nicht auf. Und unsere englischen Forth-Freunde, habe ich das Gefühl, freuen sich riesig, daß sich da was Internationales tut.)

2 Forth News

Dave Abrahams <d.j.abrahams@cwcom.net>

Das HC11-Hardware-Projekt (Jeremy Fowell) ist nahezu auslieferungsbereit. Dave Pochin hat seine Getting-Started-Webseite auf den neuesten Stand (entspricht Version 4.1 von Win32Forth) gebracht: www.sunterr.demon.co.uk/guide.htm. An der Harvard-Universität wird jemand gesucht, der ein Tunneleffekt-Mikroskop in Forth und C programmieren kann. Richard C. Wagner ist neuer Direktor und Tagungsleiter der FORML-Konferenz: 19.-21. November 1999. 10 Vorträge vorgesehen, darunter C.H. Ting "Was ist Forth?". Allegro-Projekt "Spiele in Forth": <http://www.freeyellow.com/members2/gx2/>. Gary Shannon möchte "Visual Forth" in Angriff nehmen und sucht Mitwirkende. Forth in Japan? Forth

in Java. 8 freie Forth-Systeme mit Download-Adressen. 5 Kommerzielle Systeme: 145 - 245 englische Pfund.

6 F11-UK, FIG UK Hardware Project Jeremy.Fowell@btinternet.com

Das lang angekündigte Projekt ist fertig: HC11 mit Pygmy-Forth. 47 englische Pfund zuzüglich Porto.

7 Forth - the Last 20 Years

Jack Brien <jackbrien@ballinamallard8.freemove.co.uk>

Der erste der beiden diesmaligen Hauptartikel. Die letzten zwanzig Jahre. Der Artikel auf Seite 14 dann über die nächsten zwanzig Jahre. Man beachte auch auf Seite 40 die von Fritz in unser aller Namen dargebotenen Glückwünsche zum Zwanzigsten und die Glückwünsche von Willem Ouwerkerk im Namen der HCC-Forth-gebruikersgroep.

"Prähistorisches" - FIG-Forth - Die ersten Standards - Forth unter einem Betriebssystem - Forth alleine - Forth und andere Sprachen - Der Einfluß der PC-Kompatiblen - Windows und danach - 9 Literaturangaben. Chris Jakeman hat in seinem Editorial recht: Dieser Artikel ist wirklich gut - und äußerst interessant. Nicht jeder kennt die "ganze Forth-Geschichte" und nicht überall kann man sie so kompakt nachlesen. Für "uns" nicht uninteressant: GForth als "Forth in C" und das "Datenbank-Forth" Holon werden, wenn auch nur ohne Namenszuweisungen, als Meilensteine erwähnt. Manche andere Forths auch, vorwiegend für Mikrocontroller. Transputer (Marcel Hendrix u.a.) scheinen da nicht dazugehört zu haben. Na ja, man kann eben nicht alles haben. Die Erwähnung von F-PC als eines der ersten nicht mehr unbedingt blockorientierten Systeme ist sehr lobenswert. Das leitete natürlich eine wiederum neue Denkrichtung ein. Aber warum wird eigentlich Turbo-Forth nicht erwähnt? Die Freunde aus Frankreich (hauptsächlich Marc Petremann) haben da seinerzeit (eher noch als F-PC) und auch heute noch ungeheure Arbeit hineingesteckt.

14 Forth - the Next 20 Years

Stephen Pelc <sfp@mpeltd.demon.co.uk>

Ich, der Rezensent, übersetze die Einleitung dieses 10 Seiten langen interessanten Beitrags des Autors, der von Anfang an "dabei war" und der einen couragierten Blick in die Kristallkugel der Forth-Zukunft wirft. Chris hat wirklich recht: Auch dieser Artikel, in seiner Art, ist ein Meisterwerk. Daran ändert sich auch dadurch nichts, daß ich das eine oder andere zu "meckern" haben werde.

"Englisch ist die Lingua Franca der gesprochenen Sprachen. Und eine solche brauchen wir. Es braucht gar nicht immer die beste Sprache zu sein, aber irgendeine Sprache muß es sein. Mit der Zeit wird sich das ändern. Früher, vor dem Englischen, hatte man Latein und Französisch. Und es gibt Stim-



men, die voraussagen, daß zukünftig Spanisch oder Chinesisch die Rolle von Englisch übernehmen könnte. Auf dem Gebiet der Programmiersprachen heißt die heutige Lingua Franca C, nicht Forth.

Forth ist eine bedeutende "Sprache von geringerer Bedeutung", auf dem Gebiet der eingebetteten Systeme und anderswo. Andere "Sprachen von geringerer Bedeutung", wie Pascal, Smalltalk und Modula 2, haben einen kleineren Marktanteil als Forth. In diesem Zusammenhang möchte ich mir ein paar Gedanken über die geschichtliche Entwicklung von Forth machen, um ein paar Voraussagen über dessen Zukunft wagen zu können."

Soweit der Autor. Der Rezensent möchte hinzufügen: Eine Lingua Franca ist gut, eine Verkehrssprache, ganz egal welche. Aber wenn ganze Erdteile diese Lingua Franca als Muttersprache sprechen, ist das Gleichgewicht schon etwas gestört. Eine Lingua Franca ist als Zweitsprache eine Hilfsprache und sollte von allen Beteiligten gleichermaßen (mehr schlecht als recht) aktiv beherrscht werden (können). So wie damals bei Latein. Passiv hätte man es gern gesehen, wenn auch die Sprachen der anderen berücksichtigt werden würden. Ich habe an der TU-München jahrelang mit verlässlicher Regelmäßigkeit Hochglanzwerbeproschüren einer bestimmten Forth-Firma aus Deutschland ins Institut geschickt bekommen. In Denglisch, einer anderen Art von Lingua Franca, mit vielen Auslassungszeichen und keinem Bindestrich. Dem vorliegenden Bericht entnehme ich, daß diese Firma keinerlei Einfluß auf das Forth-Geschehen hatte. Interessant! Andererseits habe ich von den im Bericht genannten Firmen nie Prospekte bekommen. Schade. Ich hätte ja als Multiplikator wirken können. Buchverlage wußten das. Eine Firma aus deutschen Landen gibt es, die erwähnt wird: Klaus Schleisiek M2CIX1-Prozessor wird als wesentlicher Hardware-Beitrag gewürdigt.

Abgesehen davon, daß jeder Autor die Dinge aus seiner Sicht sieht, ist dieser Artikel ausgezeichnet! Ich zitiere weiter aus dem Inhalt: Forth ist keine Religion - Von der Funktion her gesehen, ist nichts an Forth schlecht - Forth-Programmierer erfinden das Rad gern immer wieder neu - Ich-auch-Compiler fördern nicht, sie stören - In C schreibt keiner Compiler; Programmbibliotheken braucht die Menschheit ...

Der Autor gibt eine extrem gute Übersicht über seine Welt der Forth-Vertreiber (ich meine im Sinne von "vendor"). Reservierte Haltung gegenüber Public Domain, mit Zugeständnissen an F83, F-PC und WinFor32 - mit guten Worten für GNU und Co. Reservierte Haltung gegenüber Standardisierungen.

Der "PC als Lingua Franca der Hardware". "C++ als Lingua Franca der OOP". Und hier kommt's: "Software-ICs" als wünschenswerte Entwicklung der Zukunft. Nachtigall, ik hör dir trapsen! Das wär doch mal wieder was zum Mitmachen ??? Damals, vor der Einführung der Maikäfer, hatte man ja auch eine ganze Weile gebraucht.

"Gute Bücher und Einführungen in Forth beeinflussen die Gesellschaft weit mehr als das Schreiben von noch so vielen Compilern." Anton Ertls RAFT-Projekt wird erwähnt, verbunden mit ein paar Voraussagen über die Zukunft der Hardware.

Zusammenfassung des Rezensenten: Ich bin überwältigt und hin- und hergerissen. Ich bin sicher, dieser Artikel wird noch oft zitiert werden. Wie jede zukunftsweisende Bündelung von Gedanken, läßt aber auch dieser Artikel viel Spielraum für Rede und Gegenrede. Einem Menschen, der durchaus vorgebildet sein darf, der aber nur mal so aus Spaß an der Freude mit Forth herumspielen möchte, um seine eigene Kreativität auszuleben, ohne sich gleich mit Chomsky-Sprachen und dem Wortproblem auseinandersetzen zu müssen, würde ich diesen Artikel nicht als Eingangslektüre empfehlen. Ein solcher Mensch wird von den hier vorgetragenen verkaufsstrategischen Forth-Philosophien einfach plattgedrückt. Er wird sich von einem derartigen Schlag an verkaufsorientierter Professionalität nicht erholen.

24 20th Anniversary Reunion

Chris Jakeman <cjakeman@bigfoot.com>

FIG UK wurde im November 1979 geboren. Erster 1. Vorsitzender: Bill Powell. Gefeierte wird (wurde) am 13. November 1999 im Royal Westminster Hotel (an der Victoria Station) in London. Abgedruckt wurde spaßeshalber die Speisekarte. Interessant, die Preise mit den unseren zu vergleichen.

27 Radio Clock

Mike Trueblood <>trueblood_m@compuserve.com>

Das ist ein Artikel, den der Autor in ähnlicher Form schon 1994 in Electronics Today veröffentlicht hat. Enthalten ist eine Beschreibung des aus Rugby kommenden Zeitsignals. Letzteres ist auch zu finden unter: www.npl.co.uk/ctm. Das Listing enthält 19 Colon-Definitionen. Seltsam, keine einzige CODE-Definition. How come? And how möglich? Das vorgetragene Projekt möchte der Industrie keine Konkurrenz machen. Reiner Spaß an der Freude.

34 Clock Challenge

Chris Jakeman <cjakeman@bigfoot.com>

Gesucht: Forth für ein eingebettetes System, das das Funkuhrsignal aus Rugby aufnimmt und nach bestimmten Vorgaben reagiert. Mit dem FIGUK-Hardwareprojekt (nahezu vollendet) wurde der Grundstein gelegt. Chris: "Wir suchen Projektteilnehmer, die bereit sind, ihre Fehler, Irrwege und Erfolge mit anderen zu teilen, mit dem Ziel, ihre eigenen Kenntnisse und Fähigkeiten und die der anderen zu erweitern."

36 Dutch Forth Users Group

Diesmal wieder die Anzeige der HCC-Forth-gebruikersgroep zur Mitgliederwerbung und allgemeinen Reklame. Das nächste Mal sind wir wieder dran.



37 FIG UK Annual General Meeting

Bericht von der Jahresversammlung. 105 Mitglieder. Web site. IRC usw. Allerhand Neues. Eine Menge Bewährtes.

38 The Swap Dragon - Mascot and Annual Award of the German Forth Interest Group

Fred Behringer <behringe@mathematik.tu-muenchen.de>

Klappern gehört zum Handwerk! Mit Bild! Vom Drachen. Zu gegebener Zeit mehr!

40 Vierte Dimension

Alan Wenham

Alan hat wieder über unsere VD berichtet; zwar nur ganz kurz, aber immerhin. Danke, Alan! Diesmal über die Hefte 2/99 und 3/99. Einen Vorschlag hätte ich zu machen, Alan: Könnten Sie nicht zu den Autoren auch gleich die E-Mail-Adressen mit angeben? Das würde sicher manches vereinfachen. Ich weiß beispielsweise, daß Graeme Dunbar und Martin Bitter aus ähnlichem Anlaß E-Mails ausgetauscht haben. Heutzutage ist das ja so einfach. Große Höflichkeitsfloskeln und lange Vorstellungsriten werden nicht mehr benötigt, um schnell mal eine "Message rüberzuschicken" - "rüberwachsen zu lassen", wie ich im Fernsehen und in der S-Bahn ständig höre.

FORTHWRITE der FIG UK, Großbritannien

Nr. 105 Januar 2000

Chris Jakeman begrüßt in seinem Editorial vier neue Mitglieder, geht kurz auf den Inhalt des vorliegenden Heftes ein und streicht insbesondere Alan Wenham in seinen Bemühungen heraus, den Lesern der Forthwrite einen "expanded report on Vierte Dimension" zu bringen.

2 Forth News

Dave Abrahams <d.j.abrahams@cwcom.net>

Bericht über EuroFORTH 99 und "Machine Forth for the ARM processor" von Reuben Thomas; Forth und eingebettete System im "Electronic Design Online magazin"; die holländische Website; Links von einer Website zur anderen über den "Web Ring".

4 F11-UK, FIG UK Hardware Project

Jeremy Fowell <Jeremy.Fowell@btinternet.com>

Diesmal geht es um das FLASH-Memory für den HC11. Pygmy-Forth oder nicht Pygmy-Forth für dessen Betriebsprogramm, war hier die Frage. Aber FLASH ist natürlich ein wesentlicher Fortschritt gegenüber EPROM, und das Projekt nimmt Gestalt an. Könnten wir nicht ab und zu von unseren

eigenen Projekten in der VD Ähnliches lesen?

6 Did you Know - EasyWriter

Chris Jakeman <cjakeman@bigfoot.com>

Interessantes von früher und heute. "Wußten Sie schon", daß EasyWriter im Gefängnis geschrieben wurde? John Draper, der vorher schon Forth für den Apple II implementiert hatte, war dabei erwischt worden, wie er kostenlos nach Übersee telephonierte. Hinreichend oft natürlich, so daß eine saftige Gefängnisstrafe dabei herausprang. Er brauchte einen kostengünstigen Editor für seine Forth-Dokumentationen. In der Gefängniszelle schrieb er, als Freigänger probierte er es aus. Es gelang ihm dann, mit IBM Kontakt aufzunehmen. EasyWriter kam gerade rechtzeitig mit dem ersten PC und für diesen am 12. August 1981 heraus.

9 FIGUK 20th Anniversary Reunion

Chris Jakeman <cjakeman@bigfoot.com>

Etwa 20 Mitglieder waren gekommen. Das weitestgereiste und gleichzeitig jüngsteingetretene war Federico de Ceballos. Er war auf der Durchreise nach Kalifornien, wo er auf der FORML 99 einen Vortrag zu halten hatte. (Der Rezensent: Federico ist Mitglied des WebForth-Teams, wo er uns den spanischen Teil besorgte.)

10 Clock Challenge

Chris Jakeman <cjakeman@bigfoot.com>

Neues vom Uhren-Hardware-Projekt. Die Teile gibt es, wie schon zu Zeiten des VC20, bei Maplin im Versand: www.maplin.co.uk . Das Funkuhr-Signal kommt aus Rugby. "Die deutschen Forth-Freunde verwenden besser den deutschen Funkuhrdienst. Die IRC-Diskussionen waren rege. Vielleicht wird ein internationales Projekt daraus?" (Der Rezensent: Das waren die Worte von Chris Jakeman. Ich befand mich selbst in der IRC-Sitzung mit reger deutscher Beteiligung. Interessant, wer sich da alles von uns einfindet und wieviel da erledigt wird. Aber wo bleiben die Berichte über wirklich durchgeführte Projekte bei uns?)

12 Vierte Dimension 4/99

Alan Wenham <101745.3615@compuserve.com>

Hervorragend, wie der Alan das hingekriegt hat! Mit unserem neuen Logo (oder ist es das noch nicht?). Okay, ich arbeite mit ihm zusammen, wie er in einem Brief (veröffentlicht) an Chris Jakeman voller Dank erwähnt, aber das ist es ja gerade, was bei dieser Arbeit Spaß macht: Die Zusammenarbeit. Wir haben also einen Weg gefunden, Alans kurzzeitige Verzagt-heit auszuräumen.

14 Deutsche Forth-Gesellschaft

Und hier sind wir wieder dran, mit allen Adreßanpassungen und allem Drum und Dran. Natürlich wollen wir keinen eng-



lischen Forth-Freund zwingen, Deutsch zu lernen. Aber ein bißchen Aufmerksamkeit auf uns lenken, schadet ganz bestimmt nicht. Und Chris Jakeman bringt unsere Anzeige in wirklich anspruchsvoller Form. Danke Chris!

15 "(SC)roll Up, (sc)roll Up, see Win32Forth scroll the Window."

Dave Pochin <davep@sunterr.demon.co.uk>

Der Autor erforscht Windows über Win32Forth. Das ist ein weiterer einer Reihe von Artikeln von ihm zum Thema. Diesmal geht es um den Einbau von Windows-artigen Fenstern in das eigene Anwendungsprogramm. Vier Seiten Win32Forth-Programm. Der Fensterinhalt kann natürlich horizontal und vertikal per Mausclickundzug verschoben werden. Eigenartig, solche Fenster gab es schon 1988 bei den ersten Schritten von Turbo-Forth massenweise, von verschiedenen Autoren, in immer neuer Form. Heute müssen sie aber offensichtlich dem MS-Windows-Aussehen nachgeäfft sein, und mit dem von Windows gelieferten, völlig undurchsichtigen Material erzeugt. "Es sind nur drei Windows-Prozeduren, die man dazu braucht", sagt der Autor. Sind das aber nicht doch genau drei zuviel?

22 Forth for Fun Autorenkollektiv

Wir wissen inzwischen, was drinsteht in diesem Gemeinschaftsartikel über 7 Seiten von 6 Autoren aus 4 Ländern in 2 Kontinenten (und 1er Insel): Wort für Wort dasselbe wie in der VD. War gar nicht so leicht, die Gemüter zusammenzuführen und die Bestrebungen zu kanalisieren. Natürliche Sprachen sind offene Sprachen. Forth ist es auch. In den natürlichen Sprachen sind die Schriftsteller und Schreiber stolz auf ihre dichterischen Freiheiten. In Forth wollen wir das auch sein dürfen. Professionalismus ist gut. Aber nicht jeder, der schreibt, kann Schriftsteller sein. Trotzdem, schreiben tun wir alle. Manchmal einfach nur so aus Spaß. Oder zum Üben. Warum soll es in Forth anders sein - fragen die sechs Autoren. Apropos Üben: Englisch kann jeder, aber wenn es darauf ankommt, für die VD etwas zu übersetzen, finden sich nur wenige. Hier ist Gelegenheit, zu üben - und den Übersetzer und seine englischen und amerikanischen Helfer zu überprüfen.

29 Letters

Ein Brief (an den Editor) von Bill Stoddart über "reversible programming". Beispiel: GeDROpten Wert auf einen History-Stack legen. Wenn das Programm in eine Sackgasse geraten ist, Wert wieder hervorholen und Schritt für Schritt im Programm zurücklaufen. (Der Rezensent: Irgendwie erinnert mich das an die Markierungsalgorithmen zur Lösung von Irrgartenproblemen. Ich werde mir die reversible Programmierung als weiteres Studienobjekt vormerken. Eigentlich wäre das etwas für eine Arbeitsgruppe (Projekt-Team)!

Desweiteren ein längerer Brief von Gordon Charlton über die an ihn gestellte Bitte des Editors, seine seit fünf Jahren unterbrochene Aktivität in Sachen Forth wieder aufzunehmen, und über die Frage, die er sich selbst stellt, nämlich die, wozu er eigentlich als Hobbyist in der modernen Welt noch Forth braucht. "Etwa, um sich", so seine Worte, "wie Bill Gates, eine vollautomatisierte Wohnung einzurichten?" (Der Rezensent: Die Frage kommt mir sehr bekannt vor. Hobbyist sein, bedeutet aber doch nicht, von gar nichts eine Ahnung zu haben. Im Gegenteil. Man schleppt Probleme aus dem eigenen Fach- und Forschungsgebiet mit sich herum und überlegt sich, ob man sie nicht vielleicht mit Forth lösen kann, ohne sich deswegen gleich zum Forth-Fachprogrammierer ausbilden lassen zu müssen. Beispiel 1: Wie kann ich mich heuristisch der Frage nähern, ob es unendlich viele Primzahlzwillinge gibt? Beispiel 2: Wie schaffe ich es, das Nucleolus-Problem der n-Personen-Spieltheorie für mehr als $n = 5$ nach dem Verfahren von Behringer zu lösen (2 hoch (2 hoch n) Nebenbedingungen). Herkömmliche Sprachen scheitern an der übergroßen Zahl von Nebenbedingungen. In Forth könnte man vielleicht die ganz besondere Struktur (ganz wenige Einsen und sonst lauter Nullen) des Problems ausnutzen und den Computer "zurechtbiegen". Einem angehenden Nurmathematiker als Diplomanden konnte ich das Problem nicht geben. Einen solchen konnte man nicht zwingen, sich mit Forth auseinanderzusetzen, und schon gar nicht, einen LötKolben in die Hand zu nehmen. An Gordon Charlton gewandt: "Computer" heißt doch "Rechner". Eigenartigerweise kommt kein Mensch mehr auf den Gedanken zu rechnen. Man müßte Herrn Charlton mal fragen, welches Gebiet er beruflich bearbeitet. Wie kann man denn nur so pessimistisch über die Anwendbarkeit von Forth denken und solche destruktiven Fragen stellen! Man sollte die Frage "Was kann man mit Forth anfangen" als solche verbieten!)

35 Forthwrite Subject Index 1990-1999

Die vergangenen 10 Jahre der Forthwrite, 335 Artikel, Titel und Autoren, alphabetisch geordnet nach Sachgebieten: Algorithmen, Anwendungen, Arithmetik, Arrays, Assembler, Block-Hilfsprograsmmme, das interessante Wort, Parallelverarbeitung, Kontrollstrukturen, Datenbank, Entwurf, dynamische Daten, Editoren, Editorials, Verschlüsselung, Ausnahmen, FANSI-Projekt, Datei-Hilfsprogramme, Brüche, Zukünftiges, Graphik, Hardware, Geschichte, Humorvolles, Interface, Systeminnereien, Interpreter, Interview, Bibliothek, OOF, Permutation, Darstellung, Wahrscheinlichkeit, Veröffentlichungen, Rätsel, Zufallszahlen, Reviews, Wurzeln, Suchverfahren, Mengen, Sortieren, Stacks, Standards, Zustandsautomaten, Strings, Strukturen, Systemfragen, Hilfsprograsmmme, Tutorials, Vektorisierung.

42 FIG UK Services to Members

Zeitschrift, Bibliothek, Homepage, IRC, Mitglieder, "Wir und die anderen". Jeweils 6 Seiten Beschreibung.



VIJGEBLAADJE der HCC Forth-gebruikersgroep, Nederlande

Nr. 16, Oktober 1999

Een microcontroller network Willem Ouwerkerk

"Ein Netzwerk besteht aus mindestens zwei Computern oder computer-ähnlichen Einheiten". Der Autor geht allgemein auf Protokollfragen ein und beschreibt in groben Zügen seinen Roboter, den "großen gemeinen Schnüffler". Das "Netzwerk" kommt mit 4 Drähten (I2C-Bus) aus (+24V, Null, TXD, RXD) und verbindet 20 Einheiten (11 Sensoren, 2 Motoren, 5 Lampen, ein Signalhorn und eine Spracheinheit). - Sieht aus, als ob das der Anfang einer weitergehenden Beschreibung, die da noch kommen soll, ist.

Forth-Gesellschaft e.V.

Unsere Anzeige zur Mitgliederwerbung steht wieder mal im Vijgeblaadje. Danke, Willem!

CATCH en THROW (I) De Schoolmeester

Wieder der Schulmeister! Ich habe mich da das letzte Mal wohl geirrt. Dies hier ist ein glänzender Lehr-Dialog (Rede - Gegenrede - erklärende Gegengegenrede) in der Art der Unterhaltung zwischen Achilles und der Schildkröte in Hofstadters Buch "Gödel, Escher, Bach". Der Gegenreder (der Schüler) wird einfach durch einen Gedankenstrich verkörpert, der Schulmeister (ich sollte wohl besser Lehrmeister sagen) gibt die Antworten ohne Gedankenstrich. Prima Idee! Sollten wir nachahmen! - Jetzt weiß ich, worin sich CATCH von EXECUTE unterscheidet. Ich werde sofort meine momentanen ANSifizierungsarbeiten mit neuem Mut wieder aufnehmen.

Ich bin jetzt überzeugt davon, daß sich hinter dem Autor De Schoolmeester der überaus aktive Vorsitzende und Redakteur der HCC-Forth-gebruikersgroep, Willem Ouwerkerk, selbst verbirgt. Ob er wohl meine Rezensionen liest? Einen Hinweis auf De Schoolmeester hat er mir jedenfalls noch nicht gegeben. Ich habe aber auch, zugegeben, nie danach gefragt. Macht ja nichts. Nicht jeder kann alles lesen. Und das ordentliche Mitglied legt das jeweils neueste Heft auf den Stack - Verzeihung, auf den Stapel, mit dem festen Vorsatz, es "gelegentlich zu lesen". Und nach dem LIFO-Prinzip, das man ja auch den (Verwaltungs)beamten nachsagt, gerät es dann immer weiter nach unten.

Schade nur, daß Willem das Feigenblättchen (het Vijgeblaadje) so ziemlich allein bestreitet! Wo sind denn die anderen? Aus der Forth Dimensions, der Forthwrite und den Newsgroups wissen wir, daß sie in beachtlicher Zahl und Stärke existieren - und fleißig veröffentlichen. Nur eben nicht im

Feigenblättchen. Als das Feigenblättchen (Vijgeblaadje) noch Feigenblatt (Vijgeblad) hieß (A4, 40 Seiten), vor drei Jahren und davor, haben sie dort noch wirklich schöne Sachen verfaßt.

Bestuursleden

Ich habe verschiedentliche Anfragen unserer Mitglieder über Artikel aus dem Vijgeblaadje bekommen (ehrlich gesagt, nur zwei). Daher hier nochmal Willems E-Mail-Adresse:

Willem Ouwerkerk <w.ouwerkerk@kader.hobby.nl> . Willem versteht ausgezeichnet Deutsch, aber selbstverständlich bestens auch Englisch (er veröffentlicht in der Forthwrite englisch). Unser E-Mail-Verkehr spielt sich so ab: Ich schreibe ihm in deutscher Sprache, er antwortet in holländischer - niederländischer - Sprache. In Rundschreiben ("geforwardete Rundmails"?), in die auch Chris Jakeman, der Redakteur der FIG UK, einbezogen ist, verwenden wir (Willem und ich) "natürlich" Englisch. Englisch als Hilfssprache, nicht zum Selbstzweck.

Forth-gg produkten

8052-ANS-Forth mit AF-Board kostet 225,- Gulden. Bestellen bei Willem Ouwerkerk.

VIJGEBLAADJE der HCC Forth-gebruikersgroep, Nederlande

Nr. 17, Dezember 1999

CATCH en THROW (II) De Schoolmeester

Willem reagiert auf meine Anfragen nicht. Ich nehme stark an, daß "De Schoolmeester" er selbst ist. Das ist der zweite Teil einer offensichtlich geplanten Serie zur Erklärung der ANS-Worte CATCH und THROW. Da "das Vijgeblaadje für ein vernünftiges Beispiel zur Darstellung der Wirkungsweise von CATCH zu klein ist", setzt er das ganze Forth-System in ein CATCH und zeigt, was passiert.

Meer over Color Forth Willem Ouwerkerk

Der Autor beschreibt das System "Color Forth" von Charles Moore. Es gibt kein ":" mehr, es gibt kein ";" mehr. Lokale Variablen gibt es auch nicht, geschachtelte Schleifen sind verpönt. Es wird mit einfachsten Mitteln gearbeitet. Der Systemzustand (Compilieren, Interpretieren) wird über Farben umgeschaltet und auf dem Bildschirm angezeigt. (Der Rezensent: Ich verfolge Color Forth nun schon seit einiger Zeit, habe aber noch nicht herausbekommen können, wie das auf dem Drucker ausgegebene Listing eigentlich die Farbfrage löst. Moore ist fürs Einfache. Bravo! Mein Drucker druckt nur schwarz. Mir reicht das. Vielleicht hole ich mir tatsächlich mal etwas mehr Informationen über Color Forth unter der an-



gegebenen Adresse: <http://www.value.net/~ut/>)

Temperatuurmeting via I2C Willem Ouwerkerk

Der LM75CIM ist ein Temperaturmeß-Chip mit Möglichkeiten der Signalweiterverarbeitung und des Einleitens von Schaltvorgängen. Der Autor beschreibt dessen Einsatz im AT51-Board, mit Signalweitergabe über den I2C-Bus, in Forth programmiert.

VIJGEBLAADJE der HCC Forth- gebruikersgroep, Niederlande

Nr. 18, Februar 2000

Heute fällt mir die Besprechung leicht: Den Inhalt kenne ich auswendig. Das Vijgeblaadje, das seit einigen Jahren nicht mehr so umfangreich ist, wie es mal war, als es noch Vijgeblad hieß, widmet sich diesmal ganz dem - ich zitiere Albert Nijhof, den Redakteur und Übersetzer:

"Het Vijgeblaadje is dit keer geheel gewijd aan het project FORTH FOR FUN, 6 bijdragen uit 4 landen. Het idee komt van Fred Behringer.

Hij heeft Hugh Aguilar (USA), Chris Jakeman (UK), Willem Ouwerkerk (NL), Friederich Prinz (D) en Martin Bitter (D) uitgenodigd om hun visie over Forth te geven. Zijn uitgangspunt:"

Und so weiter. "Uitgenodigd" klingt in unseren Ohren nach Druck. Nix da! "Eingeladen" heißt es auf Deutsch. Was folgt, ist der uns inzwischen aus der VD bekannte Artikel, der (auf Englisch) auch in der Forthwrite erschien und, so hoffen wir, in der Forth Dimensions ebenfalls erscheinen wird.

Schön, das Ganze mal in holländischer Sprache lesen zu können. Leichter als Bayerisch. Organisiert hat das bei unseren holländischen Forth-Freunden Willem Ouwerkerk, der die Idee sofort willig aufgriff und damit wesentlich zum Gelingen beitrug.

Der Inhalt selbst ist eigentlich gar nicht so interessant. Klar, daß jeder Forthler eine eigene Meinung hat. "Forthler sind Individualisten", sagt Fritz. Wesentlich ist vielmehr, daß es uns innerhalb eines nur einzigen Monats gelang, einen Konsens herzustellen zwischen sechs Forthlern ganz verschiedener Herkunft, Alter und Sprache, über alle Grenzen hinweg, einen Konsens darüber, was "man" eigentlich öffentlich vertreten soll, kann und darf. Es geht dabei gar nicht so sehr um Forth, es geht um die "Forth Community". Meinungen nicht etwa gleichschalten, nein, kanalisieren, bündeln, war das Stichwort, und dabei doch nicht müde werden, einen bestimmten Standpunkt zu vertreten.

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 20 Gulden 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 20 Gulden 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.

Windows2000

Erste Erfahrungen

Friederich Prinz
Friederich.Prinz@t-online.de

Forthgruppe Moers

Meine Großmutter hat immer gesagt: „Jung gefreit hat stets gereut“, und mein Vater hat mich immer davor gewarnt, mir ein Auto zu kaufen, bei dem die Kinderkrankheiten nicht ausgemerzt sind. Ich hätte auf beide hören sollen...

Windows2000 – das Betriebssystem für die nächsten 10 Jahre ? Immerhin soll es – so behauptet es Microsoft – die Lauf-sicherheit von NT 4 mit der Funktionalität von Windows98 vereinen. Plug&Play soll so begeisternd einfach funktionieren, wie ich das auf dem Rechner meiner Frau (Windows 98) gewohnt bin. Multimedia-Anwendungen jeglicher Art sollen unter Windows2000 – im Netz bereits zu W2K verkürzt – problemlos arbeiten. Die Stabilität des NICHT-DOS-KERNELS von NT 4, die dieses System aber auch erst nach dem Einspielen des 5. Servicepacks erreicht hatte, soll im neuen System selbstverständlich sein. Immerhin baut W2K auf eben diesem Kern auf. Und von diesem Kern bin ich im Grunde überzeugt. Tatsächlich fängt das NT gelegentlich auch übelste Fehler meinerseits (meist beim Programmieren)

relativ unproblematisch ab. Jedenfalls hat NT es bisher in keinem einzigen Fall zugelassen, daß einer der von mir verursachten Fehler das ganze System gefährdet.

Mit den verschiedenen Beta-Releases von W2K hatte ich schon experimentiert. Neben der Tatsache, daß sich das System ziemlich breit macht, auf der ‚kleinen‘ 2,5 Gbyte Platte gingen mehr als 800 Mbyte nur für W2K drauf, ist mir bei diesen ersten Probeläufen nichts Unangenehmes aufgefallen. Das hat mich letztlich dazu bewogen, mir gegen alle Warnungen doch ein Produkt ‚aus der ersten Serie‘ zuzulegen. Ich hätte auf diese Warnungen hören und zumindest das erste Servicepack abwarten sollen.

Dabei lief die erste Installation des nagelneuen Systems angenehm reibungslos ab. Die Boot-Orgien, die man von Windows95 und 98 und auch noch vom NT her kennt, gibt es bei W2K nicht mehr. Ein einziger Zwischenboot, nachdem die jungfräuliche Platte partitioniert und mit NTFS-2 formatiert war, reichte dem Installationsmanager. Während der Installation müssen – natürlich – ein paar Fragen beantwortet werden, zum Beispiel nach dem Namen des Benutzers, nach der Schlüsselnummer der Installations-CD und nach der Sprache und der Zeitzone, unter denen man zukünftig zu arbeiten gedenkt. Ansonsten läuft die Installation weitgehend ohne zusätzliche Aufforderungen an den Benutzer ab – sie dauert eben etwas lange. Das liegt allerdings daran, daß auch die von allen Debugcodes befreite (hoffentlich) Verkaufsversion sich mit 800 Mbyte auf der Festplatte breit macht. Wenn man weiß, daß sich auch ein multimediafähiges Windows95 noch mit weniger als 100 Mbyte installieren läßt, dann ist das ein kräftiger ‚Schluck aus der Pulle‘.

Nach der Installation habe ich zunächst alle Standardsoftware installiert, die ich nutze. Dazu gehört unter anderem der Publisher, mit dessen Hilfe die VDs gestaltet werden. Immerhin habe ich vor, die Ausgabe 2/2000 bereits unter W2K zu erstellen. Und natürlich gehören auch verschiedene Forthe zu dieser Installation von ‚Standards‘. Besondere Aufmerksamkeit galt dabei dem ZF, das unter NT nur mit einem Patch zum Arbeiten zu bewegen war. Aber W2K hat auch diesen Patch geschluckt, bzw. das ZF, und mich damit zunächst zufrieden gestellt.

Dann brauchte ich mein ZIP-Laufwerk.

W2K baut auf dem Kernel von NT 4 auf. Prima, dachte ich, dann werden die Systemtreiber von NT 4 wohl funktionieren. Ich habe mir das zum NT jüngste Softwarepaket von IOMEGA geschnappt (liegt immer auf einer CD parat) und installiert. Ergebnis: TOTALABSTURZ !

Das ganze System hing dermaßen fest, daß auch die während des Bootens angebotenen Reparatur- und Dienstroutinen nicht helfen konnten. Um eine zeitaufwendige Neuinstallation zu vermeiden, habe ich Stunden mit Reparaturversuchen verbracht; Stunden, in denen ich die doch nicht zu umgehende

Neuinstallation längst hinter mich gebracht hätte. So ein Ärger. Warum läßt es ein angeblich so sicheres System zu, daß ein Gerätetreiber installiert wird, der für dieses System völlig ungeeignet ist ? Und warum hat W2K mich nicht wenigstens vor der Installation gewarnt ? Das tut es nämlich, wenn es bei einem Installationsversuch auf einen nicht ausdrücklich von Microsoft zertifizierten Treiber stößt.

Aber es half nichts, der zweite Anlauf war unvermeidbar. Einige Stunden später war ich auf dem Installationsstand vor dem IOMEGA-Treiber. Dieses Mal habe ich vorsichtshalber bei IOMEGA nachgesehen, ob da nicht vielleicht irgendwelche Warnhinweise auf deren Server zu finden sind. Solche Hinweise gibt es nicht. Dafür bietet IOMEGA ein Treiberpaket für alle IOMEGA-Produkte an, die unter W2K arbeiten sollen (ZIP, JAZZ). Das Paket liegt zum Herunterladen bereit – 15,5 Mbyte ‚dick‘. Das treibt der Telekom Freudentränen in die Augen. Mir half allerdings der erneut aufsteigende Ärger gar nichts. Das Paket mußte her. Und es funktioniert. Mein ZIP-Laufwerk spielt jetzt auch unter W2K mit.

Um aus dem ‚alten‘ System Datenmengen deutlich oberhalb der 100 Mbyte einer (recht langsamen) ZIP-Diskette auf das neue System zu retten, bot sich das CD-ROM Laufwerk an, bzw. der CD-Brenner. Pech gehabt ! Der Brenner tut es auch nicht ! W2K erkennt das Gerät als ganz normales CD-ROM. Aber wer braucht schon zwei CD-ROMs in einem Rechner ? Ich wollte meinen Brenner wiederhaben.

Wo fängt man mit der Fehlersuche am besten an ? Ich habe zunächst mit verschiedener Brennersoftware experimentiert. WinOnCD 3.5 und 3.6 von CeQuadrat tun es nicht. Adaptec's EasyCD 3.5A und CDDirect tun es auch nicht. Diverse, auf einfache Brenngeschwindigkeit eingeschränkte Sharewaretools wollen den Brenner ebenfalls nicht ansprechen. Dann wurde mir allmählich klar, daß Windows2000 ein weiteres Treiberproblem hat.

Ich habe in meinem PC eine SCSI-Karte von Tekram stecken. Die arbeitet mit einem Standardchip von Symbios, dem DC-310/U. Symbios bieten auf ihrem Server keine speziellen W2K-Treiber für diesen Chip an, sondern gehen vielmehr davon aus, daß dieses System den Chip problemlos erkennt. Tekram sagt sogar ausdrücklich, daß man W2K ruhig machen lassen und den angebotenen Standardtreiber verwenden soll. Aber so funktioniert es eben nicht.

Versuche, die mit der Tekram-Karte mitgelieferten Treiber nachträglich zu installieren, schlagen fehl. W2K meldet, daß in der Installations-Informationsdatei (OEMSETUP.INF) keine Sektion über das Setup enthalten sei. Das ist Unfug, definitiv.

Ich bin auf die Idee verfallen, beim ‚älteren Bruder‘ nachzusehen, mir also vom NT 4 zeigen zu lassen, welche Datei dort eingebunden wird. Siehe da: dort wird eine SYMC8XX.SYS angezogen, die sich auch auf meiner Tekram-Diskette befindet. Wenn W2K nicht mit friedlichen Mitteln zu bewegen ist, diese Datei zu schlucken, dann denke ich mir, daß es auch andere Wege gibt...

Im Verzeichnis WINNT\SYSTEM32\DRIVERS befindet sich eine Datei mit dem Namen SYM810.SYS. Die wird vom Gerätemanager als diejenige Datei bezeichnet, die meinen Brenner steuert. Einfach: Ich habe die Datei von der Diskette umbenannt und die Datei auf der Festplatte durch die Datei auf der Diskette ersetzt – ergebnislos ! Nach dem nächsten Reboot hat W2K meine Änderung rückgängig gemacht. ...da existiert wohl ein Mechanismus, der alle eingebundenen Treiber bei jedem Neustart neu einspielt. Naja, das erklärt, warum das Booten mittlerweile schneller abläuft als unter NT, aber immer noch sehr viel mehr Zeit benötigt als unter Windows98... Aber W2K muß die ‚frischen‘ Dateien irgendwo herholen. Und das tut es aus einem gepackten File heraus, einer Cabinetdatei, die unter WINNT\DRIVER Cache\i386\drivers.cab zu finden ist. In diesem Kabinett(stückchen) findet sich nicht nur SYM810.SYS, sondern auch SYMC8XX.SYS ;-). In dieses versteckte, mit dem Attribut SYSTEM gekennzeichnete und schreibgeschützte Kabinett etwas hinein zu schmuggeln, ist kein großes Kunststück. Ich habe im Kabinett SYM810.SYS einfach mit SYMC8XX.SYS überschrieben und meinen Rechner neu gebootet. Ergebnis: TOTALABSTURZ !

Dieser Absturz war nicht weniger gründlich als jener mit den IOMEGA Versuchen ! Bei Bootversuchen im abgesicherten Modus ließ sich prima verfolgen, wie der von mir gewünschte Treiber SYMC8XX.SYS endlich in das System eingebunden wurde, aber dafür war dieses Mal ‚ein paar Treiber weiter‘, bei dem Versuch den AGP Bus anzusprechen, endgültig Schluß.

Ab diesem Zeitpunkt lag W2K erst einmal ‚auf Eis‘. Es hatte mich viel Zeit gekostet und jede Menge Ärger verursacht. Ohne Brenner, den ich unter anderem zur Sicherung der umfangreichen Publisherdateien benötige, kann ich das System nicht gebrauchen.

Abendliche Streifzüge im Internet haben mich dann aber doch auch auf die Server von Adaptec und CeQuadrat geführt. Adaptec schreibt, daß ab der Versionsnummern 4.xx die Brennersoftware halbwegs mit W2K zusammenarbeitet. Eine neue, W2K-taugliche Brennersoftware und ein ordentliches Update für ältere Versionen wird für Ende März angekündigt. CeQuadrat bietet seine neueste Version WinOnCD 3.7 an. Nur 119,- DM kostet die CD. Ein Schelm, wer Böses dabei denkt...

Immerhin bietet CeQuadrat seit einigen Tagen auch ein Update auf W2K für WinOnCD 3.6 an. Das ist gerade einmal ein Zehntel IOMEGA groß, also gute 1,5 Mbyte. Aber auch das wird die Telekom freuen ! Und wer die Version 3.6 nicht sein Eigen nennt, kann selbstverständlich ein Update von 3.5 auf 3.6 ‚downloaden‘. ...ob die Telekom Aktienanteile oder sonstige Rechte an CeQuadrat hat ?

Diesen einen Versuch wollte ich noch machen. Ich habe mir das W2K Update auf WinOnCD 3.6 vom Server geladen, W2K völlig neu installiert (mittlerweile mehr als 10 Mal !) und aktuell arbeitet nicht nur das ZIP-Laufwerk einwandfrei, sondern auch der Brenner (Yamaha 4416S) !

Jetzt ist erst einmal Schluß mit den Experimenten. Mit dieser Konfiguration soll die nächste VD erstellt werden – noch vor dem nächsten Jahreswechsel !

Fazit: Mit der vielgerühmten Sicherheit kann es nicht so weit her sein, wenn das System es zuläßt, daß ‚unsichere‘ Treiber ohne Warnungen installiert werden. Das Standardsoftware zu Standardgeräten nach einem Releasewechsel nicht funktioniert, läßt vermuten, daß die Neuerungen (Verschlimmbesserungen) im Kernel tiefgreifender sind, als ich bisher dachte. Die Erfahrung lehrt, daß hier gegenüber Microsoft höchstes Mißtrauen angesagt ist ! Ebenso mißtrauisch macht mich die Tatsache, daß auch die Marktführer, zu denen CeQuadrat und Adaptec sicher gehören, nicht fähig waren, ihren Kunden rechtzeitig (!) korrekte Treiber und entsprechende Hinweise zum W2K anzubieten. Die Zeit, die ich mit Installationen, Fehlersuche und Softwaresuche verbracht habe, investiert sicher nicht jedermann, wozu auch...

Ich erwarte jetzt sehnsüchtig das erste offizielle Servicepack. Zwar halte ich die gerüchteweise kursierende Zahl von mehr als 28.000 bisher entdeckten ‚Bugs‘ in W2K für einen bösen Scherz der LINUX-Gemeinde, aber empfehlen mag ich den Umstieg auf W2K nicht nur wegen des abenteuerlichen Preises zur Zeit ganz gewiß nicht.

Nachtrag: W2K soll ausgesprochen spielefreundlich sein, so wie Windows98. Ich werde DAS aktuell bestimmt nicht ausprobieren, sondern die Rennen im Formel 1 Grand Prix lieber unter Windows98 vorwegnehmen (Friederich Schuhmacher gewinnt dabei übrigens beinahe immer ;-)). Aber die Moorhühner lassen sich auch unter W2K ‚bejagen‘ ;-)

Der verschlüsselte Schlüssel oder X S 2XOR

Fred Behringer
<behringe@mathematik.tu-muenchen.de>
Planegger Str. 24, 81241 München

Seit dem interessanten Vortrag von Ulrich Hoffmann auf der von Friederich Prinz bestens organisierten Forth-Tagung 1998 in Neukirchen-Vluyn wissen wir, wie heutzutage verschlüsselt wird. Seit dem von "yours truly" übersetzten Artikel von Hugh Aguilar in der VD 1/00 wissen wir, wie Caesar vor 2000 Jahren, wie Leon Battista Alberti vor 400 Jahren und wie die Militärs in der Zeit des Amerikanischen Bürgerkriegs das machten.

Seit dem Glühlampenrätsel von Georg Beierlein in der VD



3/99 ist bei uns das Rätselfieber ausgebrochen. Ich möchte diesen Trend fördern und stelle Ihnen heute die weiter unten umrissene vierteilige [(1),(2),(3),(4)] Aufgabe.

Zunächst die Einstimmung. Am Postbankschalter muß ich bei Barabhebungen meine PIN-Nummer eingeben. Ich habe drei Versuche, bevor die Euro-Card einbehalten wird und eingeschickt werden muß. Beim Homebanking muß ich neben der PIN-Nummer auch die TAN-Nummer bereithalten. Der Automarder muß nach Ausbau meines Autoradios bei sich zu Hause meine Keycode-Nummer eingeben, bevor er sich seines Diebstahls freuen kann. Auch der hat nur drei Versuche. Er war seit Einführung der Keycodes auch nie wieder mit der Brechstange unterwegs.

Jetzt die Anweisung. Schicken Sie mir Ihre "persönliche Geheimzahl" X in verschlüsselter Form. Verwenden Sie als Schlüssel S eine 32-Bit-Zahl, geben Sie auch X als 32-Bit-Zahl vor, halten Sie beides als 2CONSTANTen bereit und bilden Sie die mir zu übermittelnde verschlüsselte Geheimzahl Z , indem Sie die Verknüpfung $X \oplus S$ vornehmen. (Sie arbeiten mit einem 16-Bit-Forth und haben kein \oplus ? Bauen Sie sich gefälligst eins. Die Leichtigkeit, mit der das getan werden kann, macht uns ja so stolz auf unsere Sprache Forth.)

Ich verschlüssele Z nach dem gleichen Verfahren mit meinem Schlüssel T , indem ich $Z \oplus T$ bilde. Das Ergebnis W schicke ich Ihnen wieder zu. Sie verschlüsseln W abermals mit Ihrem Schlüssel S , indem Sie $W \oplus S$ bilden, und schicken mir das Ergebnis zu.

Ein Code-Spion (Abwehrdienst) hat keine Chance. Die hin- und hergeleiteten Botschaften sind immer verschlüsselt und es gibt vier Milliarden Möglichkeiten des Ratens.

(1) Beim zweiten Empfang bin ich in der Lage, die "Geheimzahl" im "Klartext" zu lesen. Nach welchem Prinzip?

(2) Nach dem zweiten Empfang kenne ich auch Ihren Schlüssel. Wieso?

Sie brauchen mir zukünftige Nachrichten also nur je einmal statt zweimal zu senden.

(3) Der Mann (die Frau) am "Message-Scanner" im Abschirmdienst schafft es, die Geheimzahl zu knacken. Der 13jährige Hacker an Omas PC auch. Wie?

(4) Wieviele Nachrichten müssen abgefangen werden, um die Geheimzahl herauszubekommen?

Krypto-Experten unter uns sind von der Bewertung der eingehenden Antworten ausgenommen. Die haben an anderem Platz eine wichtigere Funktion.

Fred Behringer

Rätsel

Fred Behringer
 <behringe@mathematik.tu-muenchen.de>
 Planegger Str. 24, 81241 München

Die folgende Aufgabe stammt von Henry Vinerts, unserem langjährigen Auslandskorrespondenten aus den USA. Sie wurde vor einiger Zeit in der Silicon Valley FIG heftig diskutiert. Allerdings gestand mir Henry in seiner letzten E-Mail, daß er seine Lösung verlegt habe und nicht mehr genau wisse, wie es gehe. Ich wußte zunächst auch nicht genau, wie es gehen sollte und habe daraufhin die Regeln der Aufgabe nach eigenen Gesichtspunkten zurechtgebogen. Die Aufgabe lautet:

(1) Man drücke die natürlichen Zahlen von 1 bis 50 dadurch aus, daß man die Ziffer "4" genau viermal verwendet und keine andere als die folgenden Rechenoperationen zuläßt: $+$ $-$ $*$ $/$, die Fakultät ($n! = 1*2*3*...*n$), die Quadratwurzel $\text{SQRT}()$ und das Quadrat $()^2$.

Damit ist die Aufgabe eindeutig gestellt. Zur Einstimmung auf das Problem sind aber noch ein paar Bemerkungen am Platz. Wenn man die Quadratwurzel als "natürliche" Operation zuläßt, ist es ebenso "natürlich", das Quadrat zuzulassen. Man könnte, was hier nicht getan werden soll, das Quadrat als Potenz mit dem Exponenten $\text{SQRT}(4)$ auffassen. Damit wäre die Möglichkeit gegeben, die allgemeine Potenz zur zulässigen Operation zu deklarieren. Wie steht es aber mit der Quadratwurzel selbst? Das ist ja nur ein ganz spezieller Fall einer Wurzelbildung. Man könnte die Quadratwurzel wie folgt schreiben: "Zweite Wurzel" = " $\text{SQRT}(4)$ -te Wurzel". Das wäre aber eine Definition, die sich selbst enthält. So, wie man sie bei manchen überschlauen Begriffen in den erklärenden Wörterbüchern findet. So etwas sollte man vermeiden. Ein Forthler hat damit allerdings weniger Schwierigkeiten. Er denkt ganz fest an RECURSE oder RECURSIVE, allenfalls an MYSELF, und alle Bedenken sind verflogen.

Soweit die begriffsphilosophische Betrachtung über die Sinnvollheit der Voraussetzungen und die Zweckmäßigkeit der Bezeichnungen. Zurück zur Aufgabe (1). In (1) sollen die Quadratwurzel und die Quadratbildung als Operationen zugelassen werden. Eine vielleicht mögliche Verwendung der Ziffer "4" oder eines ihrer Derivate in den Bezeichnungen für diese beiden Operationen (siehe Anschnitt zuvor) soll nicht zählen. Zählen soll die Ziffer "4" nur in den Operanden. Tip:



Rätselhaftes

Man schreibe konsequent "SQRT()" für die Quadratwurzel und vielleicht "QUAD()" für die Quadratbildung.

Und nun noch ein paar Aufgaben für solche, die wissen wollen, wie man Aufgaben dadurch löst, daß man sie in ein höheres Ganzes einbettet, "embedded problem solving", "Einbettungsverfahren". (Die Sandwich-Technik des Goldschmiedemeisters, der aus einem ganz dünnen Blech eine Rosette sauber ausschneiden möchte, ist von dieser Art. Er sägt sie im zusammengepreßten Hunderterpack, eingebettet zwischen zwei Sperrholzplatten, mit der Laubsäge aus.)

- (2) Kann man mit $+$ $-$ $/$ $*$ und der Quadratwurzel auskommen?
- (3) Gibt es andere Kombinationen von zulässigen Operationen, mit denen es auch geht?
- (4) Wie sieht die kleinste Zahl von Operationen, mit denen es geht, aus?
- (5) Gibt es mehrere Operationskombinationen, deren Zahl die kleinste Zahl nicht überschreitet?
- (6) Fallen Ihnen weitere Fragen ein?

Ich muß gestehen, daß ich zur Frage (6) sehr wohl, aber zu den Fragen (2) bis (5) keine Antworten parat habe. Nur möchte ich hier deutlich machen, daß man bei der Glosse von Soeren Tiedemann in der VD 1/00 über den Mathematiker, der bei den Elefanten zunächst einmal die Existenz- und Eindeutigkeitsfrage stellt, zwar schmunzeln kann, daß das Schmunzeln aber im hiesigen Zusammenhang sicher voreilig ist. Und man sage nicht, die Punkte (2) bis (5) haben keinen praktischen Wert! Wir haben es, gerade in Forth, laufend mit Zahlen zu tun. Schließlich heißt "Computer" ja nichts weiter als "Rechner". Und gerade wir in Forth. Wer anders als wir kann mit solcher Leichtigkeit vom Dezimalsystem beispielsweise ins Siebenersystem umschalten? Und das alles völlig ohne jegliche Überlegung. Darauf sind wir ja gerade so stolz. Und das ist es ja gerade, was die anderen nicht verstehen. Denen muß man das erst erklären. Und das verzeihen sie uns nie.

Zu guter Letzt noch ein Hinweis: $4+4$ ist keine Lösung für 8. $4+4+4+4+4$ ist keine Lösung für 20. $4+4+4+8$ ist auch keine Lösung für 20.

Offensichtlich gibt es für (1) viele verschiedene Lösungsmöglichkeiten.

- (7) Man gebe die größtmögliche Zahl von Lösungsvarianten für (1) an, indem man die einzelnen Varianten unmittelbar benennt.

In die Ruhmeshalle der absoluten Rätsellöser soll derjenige eingehen, der neben (1) auch (7) löst. Von (2) bis (5) ganz zu schweigen.

Auch hier stellt sich wieder die Eindeutigkeitsfrage: Kann es mehrere "absolute Sieger" geben? Muß man in der Rätsel-Ruhmeshalle einen zweiten Ehrensockel bauen? Über die Eindeutigkeit von Elefanten kann man lächeln. Hier geht es aber um ein echtes Rätsel.

Was heißt "Rätsel"? Die ganze Mathematik ist ein Rätsel. Und die Mathematiker sind ein Volk von Rätsellösern. Man sollte sie auf den diversen Geburtstagsfeiern und im Kreise der "gebildeten" Gesellschaft nicht immer wieder sich brüstend mit dem üblen Spruch "in Mathematik hatte ich immer eine 5" empfangen und sie aus purer Scham zur Vorgabe abgefälschter Berufsbezeichnungen treiben. Mit Goethe und Schiller und Deutsch tut man das ja auch nicht, obwohl doch keiner von uns sein Deutsch so richtig beherrscht. Aber keiner würde ohne Schamröte im Gesicht zugeben, in Deutsch irgendwann in seinem Leben einmal die Note "5" gehabt zu haben.

Fred Behringer

Hallo, Friederich,



nun da wir glücklich den ersten Monat des Jahres 2000 betreten und schon verbracht haben ist es Zeit, daß ich versuche mit meinen SVFIG Berichten nachzukommen. Obwohl ich meinen Vollzeitjob am Ende des Jahres aufgegeben habe, scheint es, daß ich weniger Zeit als zuvor habe. Der Hauptgrund für meine Langsamkeit ist jedoch, daß ich meinen alten DOS-basierten Internetdienst aufgeben mußte und nun lernen muß, meine e-mail über die AOL-Verbindung meiner Frau abzuwickeln. Ich hoffe, daß dieses File korrekt 'anhängt' und ebenso versandt wird, ansonst bin ich möglicherweise gezwungen, auf einem Microsoft Windows Textprogramm schreiben zu lernen, und dann werdet Ihr von mir lange Zeit gar nichts mehr hören.

Zuerst der Bericht über den Silicon Valley Forth Tag, 18. Dezember 1999

Die Bilder, die ich Dir schickte, waren von diesem Treffen. (Um Deine Frage zu beantworten--ja, ich kann Bilder auf Nancy's Rechner anschauen, aber ich brauche Ihre Hilfe dafür.) Den Tag über zeigten sich mindestens 45 Leute und es gab über ein Dutzend individuelle Gespräche und Präsentationen -- zu viele für mich, um sie im Detail zu behandeln. Natürlich, die Hauptattraktion war, wie immer, Chuck Moore, der sich als der 'Guru des ColourForth' vorstellte und uns als 'Überlebende' begrüßte. Ich glaube, daß ich Dir über ColourForth schon vorher geschrieben habe. Es ist nun schon eine ganze Weile (wenn ich es so nennen darf) Chucks "Steckenpferd". Wie er sagt, ist es nicht das Ziel, sondern es ist ein Weg; wie üblich ist er auf seinen neuen Wegen - wieder auf der Suche nach dem wahren Minimum einer menschlichen Schnittstelle in der Informationsverarbeitung. ColourForth legt Farbcodes in die ungenutzten Zwischenräume und



weist damit die Funktion des nachfolgenden Wortes zu. So mag ROT ein neues Wort anzeigen, BLAU einen Kommentar, GRÜN ein zu compilierendes Wort, und so weiter.

Ich möchte einige von Chucks Kommentaren wiederholen, in der Hoffnung, daß ich Ihnen korrekt verstanden habe:

"Es ist wirklich ergreifend, daß wir alle noch hier sitzen und über Forth reden. Es ist nicht der Weg der Zukunft."

"Ein Standard ist nicht notwendigerweise eine gute Sache. 8- und 32-Bit Maschinen mögen unterschiedliche Standards erfordern. Es ist falsch, daß wir alle versuchen auf der PC-Plattform übereinzukommen. Das Falsche könnte für immer eingefroren werden, und es besteht die Gefahr, daß alle es auf die gleiche Weise tun."

"Die Softwareindustrie würde zusammenbrechen, falls die Leute nur das kaufen, was Sie brauchten und nicht das, was Sie wünschen."

Und noch ein paar "Sachen zum Nachdenken" von Dr. Glen Haydon, die er schon vorher am Tag äußerte:

"Das Problem in der realen Welt besteht darin, daß wir gern etwas mehr "statisch" wären. Jedoch hat ein stabiler Kern in den frühen Forth-Versionen nie existiert."

"Es gibt keine Händler, die vom Verkauf von Forth leben können. Das wahre Einkommen liegt nur in den Beratungsleistungen."

"FORML (d.h. die Forth Modification Laboratory Konferenz) wurde auf gewisse Art und Weise durch die Standardisierung gestoppt. Die letzte Konferenz hatte 22 oder 23 Teilnehmer und 7 Gäste. Vor fünfzehn bis zwanzig Jahren zog das Treffen üblicherweise 90 bis 100 Leute an. Die Rochester Konferenz nahm von 200 auf 25 ab. Der ursprüngliche 'Forth Tag' dauerte in Wahrheit zwei Tage und an die 1500 Leute kamen. Heute haben wir 40."

"Lang lebe Forth!"

Nach den vielfältigen Präsentationen zu urteilen, lebt Forth, obwohl manchmal in Verkleidung oder in sehr spezifischen individuellen Anwendungen. Es war ein Vergnügen Mike Perry zu sehen (der mit F83 berühmt wurde). Er arbeitet mit Open Firmware. Kevin Appert nutzt immer noch Forth bei Lockheed. John Peters und Henrik Thurfjell entwickeln einen Verfolger für elektrische Stromkreise. Al Mitchell baut Überwachungsgeräte für Verunreinigungen, Tester für Autoabgase. Dave Jaffe arbeitet an verschiedenen Geräten für behinderte Menschen. Skip Carter wird uns in zukünftigen Treffen erzählen, wie Forth dazu kommen kann, eine Rolle im Netz in XML-Anwendungen zu spielen.

Als zweites, der Bericht vom SVFIG Treffen, 22. Januar 2000

Nun sind wir wieder bei nur 11 Leuten zu Beginn, doppelt so viele zur Mittagszeit. Skip Carter ist außerhalb unterwegs, und niemand sonst kann über Forth mit XML berichten. Jet Thomas spricht über das "State-Smart"-Durcheinander beim Parsen von Forth-Code, offeriert Heilmittel und bekommt Anregungen aus dem Auditorium, wobei abschließend alle darin

übereinstimmen, daß es ein Durcheinander bleibt und das "Einfachheit hinzufügen" in Wirklichkeit "Konfusion entfernen" bedeutet.

Dr. Ting hat Ideen für den Palm IIIe, als "andere Seite von eForth".

Dr. Haydon kehrt mit noch mehr Vermutungen über die Zukunft von Forth zurück: "Unsere Mitgliederzahl sinkt. Mehr Applikationen werden für Forth gebraucht."

Nach dem Essen zeigt uns John Rible eine Kopie von Jonathan B. Knudsens "The Unofficial Guide to LEGO Mindstorm Robots" (Inoffizieller Führer für die LEGO Mindstorm Roboter), welches ein Kapitel über pbForth von Ralph Hempel enthält (www.hempeldesigngroup.com/lego/pfForth).

Dr. Ting präsentiert seinen Vorschlag für die Forth-Universität -- eine online Forth-Lehranstalt, in der eine Ausbildung in Forth und Fähigkeitszertifikate für einen breiteren Kreis zur Verfügung gestellt werden können -- breiter als wir es durch traditionelle Mittel erreichen können.

Zuletzt kehrten John Peters und Henrik Thurfjell mit einer Demonstration des Leitungssuchers für Gebäude zurück -- ein nettes Werkzeug das von einem Einzelnen benutzt werden kann, ohne das man irgendwelche Leitungen unterbrechen muß.

Ich habe das Gefühl, daß Forth, wenn es überlebt, im kleinen Maßstab leben wird, als Werkzeug für einige wenige Individualisten, die in den täglichen Nachrichten nur dann erwähnt werden, wenn Sie das "Unmögliche" vollbracht haben, indem Sie einige einzigartige, aber dringende Probleme gelöst haben. Wie die sogenannten "Handy-Men" im Land, für die "kein Job zu klein ist", werden sie als Mikroprozessor-Klempner, (Wolkenkratzer) Fensterputzer, Hunde- (oder Hacker-)fänger, (Schmetterlings-) Netzflicker und so weiter dienen. Ihr Status und Ihre Popularität werden nur dann steigen, wenn Leute sich nicht das leisten können, was sie wollen und anstatt alles neu zu kaufen, zurückgehen und das Vorhandene reparieren und warten müssen.

Auf Wieder-E-Mailen!

Henry

(übersetzt von Thomas Beierlein)

Daß der Mitgliederschwund in den USA noch sehr viel dramatischer ist als wir ihn hier in der BRD feststellen, hat mir Heinz Schnitter bereits berichtet. Es gibt wenig Hoffnung, diese Entwicklung aufzuhalten. Daß ‚die Menschheit‘ aus Armut den Weg zurück zu Forth findet, wird sich Niemand ernsthaft wünschen. Tatsächlich wird Forth aber auch gerade dort eingesetzt, wo das Geld für teurere Entwicklungen und Entwickler fehlt. Im Zusammenhang mit den LOGO-Robotern werden wir demnächst zu diesem Thema aus den eigenen Reihen sicher mehr hören.

Die Idee der ‚Forth-Universität‘ scheint mir zudem geeignet, Forth zumindest wieder in die allgemeine Diskussion zu bringen. Sehen Sie das auch so ?

jep



Wanted!

Rubrik. Wir möchten hiermit eine Rubrik weiterführen, die einst von Jörg Staben ins Leben gerufen wurde (VD 3/92) und laden alle Forth-Freunde dazu ein, entsprechende Ideen für ähnliche Projekte beizusteuern. Ganz besonders wären auch nichtdeutsche Zuschriften willkommen. (Schade, daß uns in unserer eigenen Sprache in England und Amerika wahrscheinlich nur drei Getreue lesen.)

Analogrechner. Voll digital. Möglichst wenig Hardware, allenfalls ganz kleine Zusätze. Weg von der "Ingenieurgenauigkeit (3%) einstiger Zeiten. Voll in die Zukunft von damals, in unsere heutige Welt der nahezu unbegrenzten Möglichkeiten (was den Geldbeutel des normalen Forth-Enthusiasten betrifft). Vielleicht den Gedanken des ehemaligen "Differential Analyzers" wieder aufnehmen. (Digitale) Potentiometer aber mit 64-Bit-Genauigkeit, Integrierer (wie am besten?), invertierende Summierer (selbstverständlich), Multiplizierer (ebenso selbstverständlich), Dividierer (verhältnismäßig selbstverständlich), Funktionsspeicher (bei 256 Megabyte RAM wirklich kein Problem mehr - was hätten wir uns damals gefreut!). Differenzierer braucht man nicht: Jede noch so komplizierte (gewöhnliche) Differentialgleichung läßt sich als System darstellen, in welchem jede einzelne mit nur einem einzigen Integrierer "integriert" werden kann. Statt "Analogrechner" natürlich sofort gleich "Hybridrechner": "Blockorientierte Sprache" (siehe ehemalige breite Literatur über Analogrechner) für den Differentialgleichungsteil (selbstverständlich in Forth, selbstverständlich in dem, was man heute OOP nennt), "normale" digitale Elemente für den Ablaufsteuerungsteil. Festkommaarithmetik (?) Gleitendes Komma ersetzen durch "Optimale Skalierung" (siehe Literatur von damals).

Wer hat Lust, bei einem solchen Projekt mitzumachen?

"Das gibt es ja alles schon!", sagen Sie? Her mit den Informationen! Selbstverständlich ist das Vorhaben nicht neu. Und wenn man auf Leonardo da Vinci zurückgreift, ist wirklich alles schon dagewesen. Wir haben aber inzwischen zig Stufen der Entwicklung erlebt und haben gestaunt. Jetzt, da wir uns vom Staunen langsam erholen, wäre es sicher angebracht, ursprüngliche Ideen wieder aufzugreifen. Mit neuesten Mitteln. Damals haben wir von einem digitalen Potentiometer mit 64-dBit-Genauigkeit nur träumen können. Heute bekommen wir es bei Aldi mit einem ganzen Computer drum herum aus der Kühlkiste. Im übrigen kupfern wir gern ab. In Forth geht es uns so wie in der Goetheschen Bemerkung über Übersetzungen und die Franzosen: Wir übernehmen einen Programmausschnitt, übertragen ihn nach Forth, und schon ist es etwas ganz anderes, etwas völlig Neues.

An sich schmilzt das ganze Projekt auf eine einzige Forderung zusammen: Gebt mir einen guten Integrierer und die Sache läuft - noch nicht ganz, aber so ziemlich. Es gibt zwei

Möglichkeiten: (1) zeitbezogen; dann muß irgendein Zeitgeber ("Timer") ran und es dürfen keine Interrupts oder sonstige Störungen ("Eventhandling") wirken, (2) wertbezogen; dann könnte man statt gewöhnlicher Differentialgleichungen auch gleich partielle ins Visier nehmen (damals, auf dem Analogrechner, war das noch ein unerreichbarer Traum).

Unser Freund Schmitt soll gesagt haben, er brauche kein Windows, er baue sich sein Windows selbst. Nun gut. Nehmen wir sein Lehrsystem (Vortrag auf der Tagung 1999) als graphische "Konsole" (heutzutage brauchen wir keinen Tektronix-Speicheroszillographen zum "Bildschirm" umzufunktionieren und benötigen keinen Plotter mit dauernd eingetrockneter Tinte für 10000 Mark) und bauen uns da einen Analogrechner hinein!

Soll man den Operationsverstärker digital simulieren? Soll man das Prinzip des ϵ -Integrierers, des Kernstücks der Analogrechnerei, beibehalten? Soll man OpAmps, die ja nur noch Pfennigware darstellen, in das Projekt integrieren und ein gemischtes Hard-Soft-Projekt aufbauen?

Das programmierbare Analog-Array MPAA020 von Motorola, über das in der Zeitschrift Elektor 6/98 berichtet wurde, kommt dem vorgeschlagenen Vorhaben von der Hardwareseite her recht nahe. Seit 1961 ersehne ich mir sowas. Das "automatische Steckbrett" (automatic patchboard), nach dem damals verzweifelt Ausschau gehalten wurde, war ja nichts anderes. Was nützt aber ein noch so schöner Bericht über "was es nicht alles gibt", wenn die zuerst breit angepriesenen Dinge dann doch nicht zu haben sind? Ich will nicht länger warten. Ich kann nicht länger warten (nochmal 40 Jahre?). Ich mache mir die Sachen einfach selbst (Zitat Schmitt) - soweit ich das kann. Was ich nicht kann, das laß ich eben sein. Es treibt mich ja keiner.

beh

Interessante WEB-Links

Michael Kemper arbeitet auch beruflich mit Forth, zeigt aber auf seiner WEB-Site die ‚spaßige‘ Seite ‚unserer‘ Sprache. Auf

<http://216.132.198.179/>

können Sie nachvollziehen, wie Michael Kemper Spiele, bzw. deren Programmierung, dazu nutzt, sich selbst über unterschiedliche (programmtechnische) Probleme und mögliche Lösungen zu informieren. Win32For und SwiftForth sind dabei seine Werkzeuge.

Michael Kemper lädt dazu ein, sich die Sourcen zu den von ihm programmierten Spielen herunter zu laden. Die von ihm definierten Klassen in Win32For und SwiftForth stehen ebenfalls frei zur Verfügung.

Schauen Sie mal vorbei. Es lohnt sich.

mkm



DO--LOOP mit Code von bis zu 32 K Länge ab 80386 für Assembler in ZF und Turbo-Forth

Fred Behringer
<behringe@mathematik.tu-muenchen.de>
Planegger Str. 24, 81241 München

Warum soll man sich in 16-Bit-Systemen mit Assembler-Sprüngen von nur 128 Bytes begnügen, wenn die CODE-Definitionen (theoretisch) bis zu 64 Kilobyte lang sein dürfen?

Stichworte: ZF, F83, Turbo-Forth, Assembler, DO--LOOP, Jxx-Sprünge über 32 K.

Das Forth-Wort LOOP im Assembler von ZF, Turbo-Forth und den anderen F83-Abkömmlingen greift, wie könnte es anders sein, auf den Intel-Befehl LOOP zurück. Dieser erlaubt Schleifensprünge zurück an den Anfang nur über eine maximale Distanz von 128 Bytes, auch ab dem 80386 bis hin zum Pentium in seinen verschiedenen Ausprägungen. Nun ist der LOOP-Befehl aber "nichts weiter" als eine Zusammenfassung aus CX DEC und einem bedingten Sprung. Bedingte Sprünge können aber ab dem 386 (auch im Real-Mode und unter ganz normalen Bedingungen) Entfernungen von bis zu 32768 Bytes überbrücken. Statt der bedingten Sprungbefehle mit dem Opcode 7x und einem 1 Byte breiten Operanden sind einfach nur die damit korrespondierenden Befehle mit dem Opcode 0F 8x und einem 2 Byte breiten Operanden zu verwenden. Wendet man das kleine Zusatzprogramm aus meinem VD-Artikel "BEGIN--UNTIL über 32 K ab 80386 im ZF-Assembler" zur Umschaltung und Ergänzung seines ZF- oder Turbo-Forth-Systems an, so ist alles gelaufen - vorausgesetzt, man fügt noch den gleich folgenden Programmteil hinzu.

Die unten aufgeführten neuen LOOP(xx)s funktionieren auch mit den bisherigen IF--THENS usw., die nur 128-Byte-Sprünge zulassen, was vielleicht für allererste Testzwecke interessant sein könnte. Natürlich steigen sie dann bei mehr als 128 Byte Schleifenlänge aus und die in die neuen LOOP(xx)s eingebaute Fehlermeldung "Sprung zu weit" wirkt nicht richtig, da sie auf 16-Bit-Sprüngen eingestellt ist.

Hat man nur ganz kurze DO-Schleifen, dann verwendet man die neuen LOOP(xx)s besser nicht. Der "Vorspann" ist doch um einige Bytes größer. Auch wenn die 16-Bit-Kontrollstrukturen (IF usw.) aus meinem VD-Artikel

"BEGIN--UNTIL über 32 K ab 80386 im ZF-Assembler" eingeschaltet sind, bleiben die DO-Schleifen auf die bisherigen Kontrollstrukturen fixiert, falls die neuen LOOP(xx)s nicht hinzugefügt wurden.

Caveat: Beim Austesten unterlief mir zuerst ein kleiner Denkfehler. Ich hatte in die drei Testprogramme jeweils vor dem LOOP 240 NOPs eingebaut, die erweiterten IF--THENS nicht eingeschaltet und mich gewundert, daß das System nicht ausstieg. Nun, bei den DO-Schleifen ist es natürlich egal, an welcher Stelle mit den NOPs der jeweils nächste Schleifendurchgang wieder einsetzt, das System bricht dadurch nicht zusammen. An den Ergebnissen (man überlege sich selbst, an welchen) ließ sich dann allerdings sofort erkennen, daß da jetzt die einzelnen Schleifendurchgänge am falschen Platz einsetzten.

CX ist das einzige Register, das von den LOOP(xx)s (auch bei den neuen DO-Schleifen) beansprucht wird. Nach vorzeitigem Abbruch der DO-Schleifen enthält es diejenige Anzahl von Schleifendurchgängen, die noch verblieben wären, wären auch die restlichen Durchgänge bis hin zum Erreichen von CX = 0 noch abgearbeitet worden. Erfolgt der Schleifenausprung erst nach Erreichen von CX = 0, dann bleibt der Wert 0 in CX natürlich auch nach Verlassen der Schleife erhalten.

Programm-Listing

HEX ASSEMBLER DEFINITIONS

```

: LOOP          \ "loop while CX <> 0"
  [ FORTH ]    \ Nimm DUP aus FORTH
  TRUE SWAP    \ Aus DO mach BEGIN
  DUP HERE - 2- 0> \ Absicherung,
                \ 16-Bit-Sprünge
  ABORT" Zuviel Code, Rücksprung zu weit"
  [ ASSEMBLER ] \ Nimm CX DEC usw
                \ aus ASSEMBLER
  CX DEC 0<> WHILE \ Abbruch, genau
                  \ wenn CX = 0
  REPEAT ;

: LOOPE         \ "loop while equal"
  [ FORTH ]    \ Nimm DUP aus FORTH
  TRUE SWAP    \ Aus DO mach BEGIN
  DUP HERE - 2- 0> \ Absicherung,
                  \ 16-Bit-Sprünge
  ABORT" Zuviel Code, Rücksprung zu weit"
  [ ASSEMBLER ] \ Nimm CX DEC usw
                \ aus ASSEMBLER
  0<> IF      CX DEC CX CX CMP
              \ <> 0 -> Zeroflag = 1
  ELSE CX DEC
              \ = 0 -> Zeroflag = 1,
              \ genau wenn CX = 0
  THEN
  0<> WHILE   \ Abbruch, wenn <> 0
              \ oder CX = 0
  REPEAT ;

```



DO—LOOP 32 k

```

: LOOPNE      \ "loop while not equal"
  [ FORTH ]   \ Nimm DUP aus FORTH
  TRUE SWAP   \ Aus DO mach BEGIN
  DUP HERE - 2- 0> \ Absicherung,
                \ 16-Bit-Sprünge
  ABORT" Zuviel Code, Rücksprung zu weit"
  [ ASSEMBLER ] \ Nimm CX DEC usw
                \ aus ASSEMBLER
  0= IF  CX DEC CX CX CMP
                \ = 0 -> Zeroflag = 1
    ELSE CX DEC\ <> 0 -> Zeroflag = 1,
                \ genau wenn CX = 0
    THEN
  0<> WHILE   \ Abbruch, wenn = 0
                \ oder CX = 0
  REPEAT ;

```

FORTH DEFINITIONS

```

CODE TEST
  AX POP 500 DO AX INC
    LOOP
      CX PUSH
    NEXT
END-CODE

```

```

CODE TESTNE
  AX POP 500 DO AX DEC
    LOOPNE
      CX PUSH
    NEXT
END-CODE

```

```

CODE TESTE
  AX POP 500 DO AX INC
    0<> IF DX DX XOR
      ELSE DX DX XOR
        DX INC
      THEN
    LOOPE
      CX PUSH
    NEXT
END-CODE

```

\ Ergebnisse

```

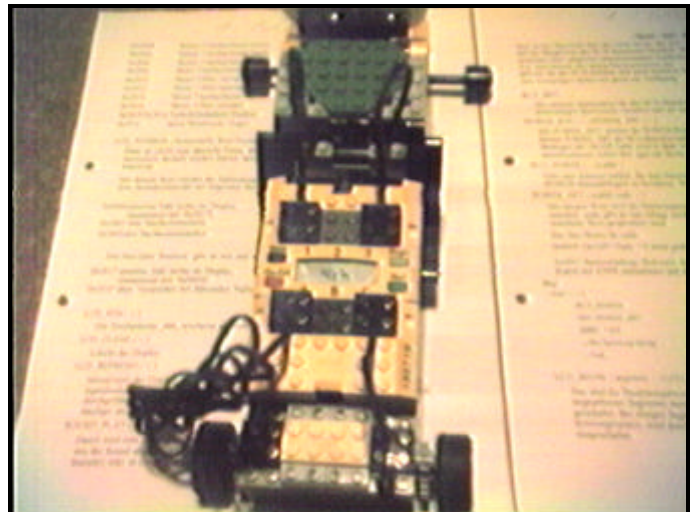
-200 TEST . \ [RET] 0 OK
           \ Schleifenausssprung
           \ ausgelöst durch CX = 0
-100 TEST . \ [RET] 0 OK CX = 0
  0 TEST . \ [RET] 0 OK CX = 0
 100 TEST . \ [RET] 0 OK CX = 0

-200 TESTE . \ [RET] 300 OK DX <> 0
-100 TESTE . \ [RET] 400 OK DX <> 0
  -1 TESTE . \ [RET] 4FF OK DX <> 0
  0 TESTE . \ [RET] 0 OK CX = 0
  1 TESTE . \ [RET] 0 OK CX = 0
 100 TESTE . \ [RET] 0 OK CX = 0

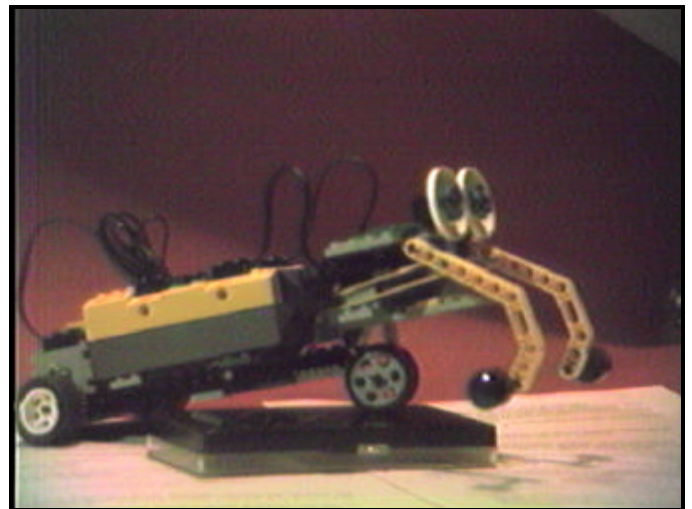
 300 TESTNE . \ [RET] 200 OK AX = 0
 400 TESTNE . \ [RET] 100 OK AX = 0
 4FF TESTNE . \ [RET] 1 OK AX = 0
 600 TESTNE . \ [RET] 0 OK CX = 0

```

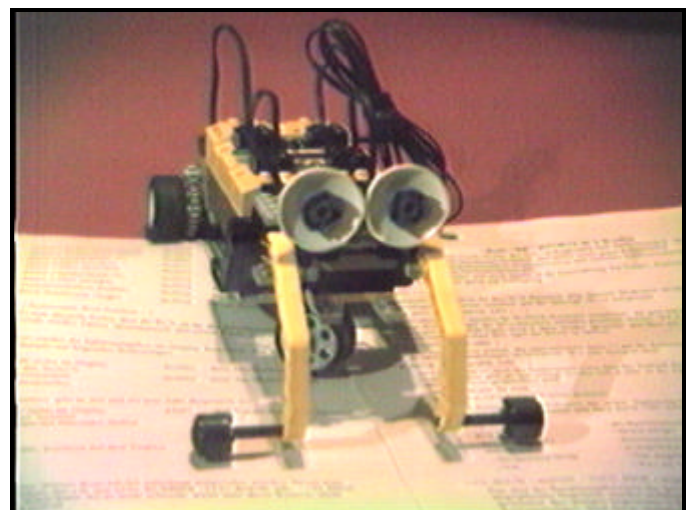
LEGO Mindstorm Robotor



Diesen ‚Robbi‘ treibt ein Forth.
Das befähigt ihn zu erstaunlichen Dingen,
von denen Sie in einer der nächsten VD
sicher mehr lesen werden.



Selbst lesen kann der kleine Kerl (noch) nicht.
Aber offensichtlich übt er das gerade.



Forth-Gruppen regional

Moers **Friederich Prinz**
Tel.: 02841-58398 (p) (Q)
(Bitte den Anrufbeantworter nutzen !)
(Besucher: Bitte anmelden !)
Treffen: (fast) jeden Samstag,
14:00 Uhr, **MALZ, Donaustraße 1**
47443 Moers

Mannheim **Thomas Prinz**
Tel.: 06271-2830 (p)
Ewald Rieger
Tel.: 06239-920 185 (p)
Treffen: jeden 1. Mittwoch im Monat
Vereinslokal Segelverein Mannheim e.V.
Flugplatz Mannheim-Neuostheim

München **Jens Wilke**
Tel.: 089-89 76 890
Treffen: jeden 4. Mittwoch im
Monat, **China Restaurant XIANG**
Morungerstraße 8
München-Pasing

mP-Controller Verleih

Thomas Prinz
Tel.: 06271-2830 (p)
micro@forth-ev.de

Gruppengründungen, Kontakte

Fachbezogen 8051 ... (Forth statt Basic, e-Forth)
Thomas Prinz
Tel.: 06271-2830 (p)

Forth-Hilfe für Ratsuchende

Forth allgemein

Jörg Plewe
Tel.: 0208-49 70 68 (p)

Jörg Staben
Tel.: 02103-24 06 09 (p)

Karl Schroer
Tel.: 02845-2 89 51 (p)

Spezielle Fachgebiete

Arbeitsgruppe MARC4 Rafael Deliano
Tel./Fax: 089-841 83 17 (p)

FORTHchips Klaus Schleisiek-Kern
(FRP 1600, RTX, Novix) Tel.: 040-375 008 03 (g)

F-PC & TCOM, Asyst Arndt Klingelberg, Consultants
(Meßtechnik), embedded akg@aachen.kbbs.org
Controller (H8/5xx// Tel.: ++32 +87 -63 09 89 pgq
TDS2020, TDS9092), (Fax -63 09 88)
Fuzzy

KI, Object Oriented Forth, Ulrich Hoffmann
Sicherheitskritische Tel.: 04351 -712 217 (p)
Systeme Fax: -712 216

Forth-Vertrieb volksFORTH / ultraFORTH
RTX / FG / Super8 / KK-FORTH
Ingenieurbüro Klaus Kohl
Tel.: 08233-3 05 24 (p)
Fax : 08233-99 71
mailorder@forth-ev.de

Forth-Mailbox (KBBS) 0431-533 98 98 (8 N 1)
Sysop Holger Petersen
hp@kbbs.org
Tel.: 0431-533 98 96 (p) bis 22:00
Fax : 0431-533 98 97
Helsinkistraße 52
24109 Kiel



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.



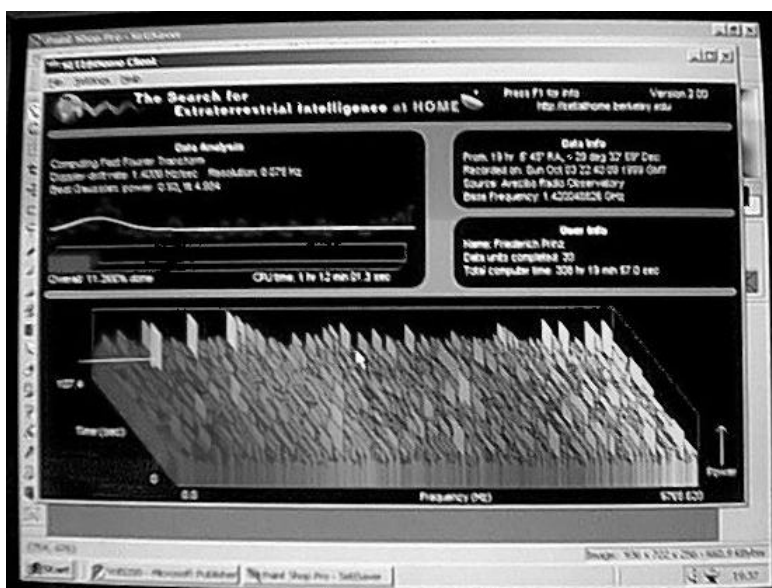
Unter welche Rubrik unserer Zeitschrift könnte das Banner des SETI-Experimentes wohl fallen ? Mit der Luft- und Raumfahrt hatte Forth eigentlich schon immer zu tun. Auf dem Mond war ‚Forth‘ natürlich auch schon. Und ein wenig lästerlich möchte ich hinzufügen, daß die allermeisten der heute populären oder früher populär gewesenen Entwicklungssprachen *hinter* dem Mond geblieben sind...

Das SETI-Experiment sucht nach außerirdischem, intelligentem Leben. Mit dem größten Radioteleskop der Welt wird auf dem 1.420 MHz Band ein relativ breiter ‚Gürtel‘ des Himmels nach Radiosignalen abgetastet. Die aufgefangenen Radiosignale müssen anschließend verarbeitet werden. Dabei fällt eine so große Menge an zu bearbeitenden Daten an, daß die dafür an der Universität von Kalifornien zur Verfügung stehenden Rechner ebensowenig ausreichen, wie die von SUN Microsystems zur Verfügung gestellte Rechnerkapazität (Sie erinnern sich? Das sind die mit dem Forth in der Boot-Firmware!). Unterstützt wird das Experiment auch von einigen anderen, namhaften Unternehmen – und von einer großen Anzahl von Menschen, die ihre privaten Rechnerkapazitäten zumindest teilweise zur Verfügung stellen.

Und das ist für mich persönlich der interessanteste Teil des Experimentes! Heute, Mitte März 2000, lassen mehr als 1,8 Millionen Menschen in der ganzen Welt ihre privaten Rechner an einzelnen Datenpaketen für SETI arbeiten. Die mittlere Verarbeitungszeit für ein einzelnes Datenpaket lag dabei bei ca. 21,5 Stunden. Mittlerweile sind in mehr als 1.870.000²⁰ Fließkommaoperationen rund 225 Tausend (!) Jahre an CPU-Zeit von diesen Menschen für die Wissenschaft eingesetzt worden!

Faszinierend – [;-)] – gibt es ein deutlicheres Beispiel für die parallele Verarbeitung von Daten? Faszinierender – es gibt wohl doch noch eine große Anzahl Menschen, die, ohne dafür irgendeinen Gegenwert erwarten zu dürfen, ihre privaten Ressourcen in ein wissenschaftliches Experiment einbringen. SETI läßt hoffen – worauf, daß mag jeder für sich selbst bestimmen.

Das Mitmachen wird bei SETI allerdings auch sehr einfach und obendrein ansprechend gemacht. Die Software für den Transfer der Datenpakete ist selbstverständlich kostenlos. Und diese Software ist ‚richtig gut‘ programmiert. Mit NT’s Task_Manager läßt sich sehr schön verfolgen, daß SETI’s Arbeitsprogramm sich an Rechnerressourcen nimmt, was immer es kriegen kann. Aber jede Anwendung auf Ihrem PC drängt SETI ein wenig weiter in den Hintergrund. Sie werden die Anwesenheit von SETI auf Ihrem Rechner gar nicht bemerken. Dafür können Sie sich, wenn Sie mögen, an einem außergewöhnlichen Bildschirmschoner erfreuen. Wenn Sie SETI erlauben – was Sie nicht müssen – als Bildschirmschoner zu arbeiten, dann bekommen Sie bei der graphischen Darstellung der aktuellen Rechenoperationen „richtig was geboten“. So etwas hat nicht Jeder!



Bleibt noch anzumerken, daß der Datentransfer das Internetkonto im Prinzip auch nicht belastet. Ein frisches, noch zu berechnendes Datenpaket ist gerade einmal 340 kByte groß. Ein solches Paket abzuholen und Stunden oder auch Tage später wieder zurück zu schicken, läßt sich bequem und kostenneutral „zwischen zwei E-Mails“ erledigen.

Habe ich Ihr Interesse wecken können ? Dann schauen Sie doch einfach einmal in die übrigens ganz hervorragend gemachten Seiten von:

www.setiathome.ssl.berkeley.edu

Viel Spaß

Friederich Prinz

P.S. Mein PII, 400 MHz, 128 Mbyte RAM braucht durchschnittlich 8,5 Stunden pro Paket !