

VIERTE DIMENSION

1/1996

12. Jahrgang 1996, 1. Quartal, DM

Forth: Profi oder Hobby?

Forth auf eine professionelle Basis heben
Forth braucht keinen qualitätsüberwachenden
Fachverband

Portables Forth: Atari-Portfolio
cpForth für OS/2
Jeff Raskins "Canon Cat"
Noch Platz ??
Das müssen wir analysieren ...
Rätsel

FORTH MAGAZIN

Organ der Forth Gesellschaft e.V.

Ingenieurbüro
Dr. Maier-Schuler

Software-Entwicklung
Prüfgerätebau
Anlagensimulation

Echtzeit-Simulation und Datenerfassung unter OS/2. Mit grafischer Oberfläche und Schnittstellen zu anderen Anwendungen.

cpFORTH

das FORTH-System für OS/2. Erweiterbar durch DLL- und REXX-Schnittstellen. Als C++ Objekt auch in User-Programme integrierbar.

Shareware- und Vollversionen verfügbar!

Dr. Maier-Schuler Wissenstraße 19 88690 Uhldingen-Mühlhofen Tel.: 07556-5618

FORTH - Shirt



T - Shirt: hellgrau / grün
in Größe: M-L-XL 25 DM

Sweat-Shirt: grau / grün
in Größe: M-L-XL 40 DM

(+ Porto)

ForthWORKS

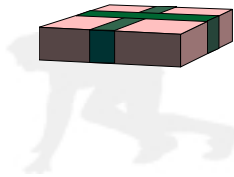
Ulrike Schnitter

Nelkenstr. 52

85716 Unterschleißheim

Tel.: 089-310 33 85

2PW



Letzte Chance...

...das Forth-Weihnachtspäckchen zu bestellen.

Die letzten CDs mit beiden Sonderdrucken gibt es gegen 45.-DM bei der Redaktion. Ausländische Bestellungen werden bevorzugt, aber nur als DM-Scheck oder bar! Übriggebliebene Bestellungen gehen an Klaus Kohl weiter, es gibt dann nur die CD.

Die CD 'forthige Weihnacht 95' enthält u.a. den kompletten Ftp-Server Taygeta aus USA mit bald 100 Mb; den kompletten Ftp-Server Asterix aus Portugal und Forth-Systeme und -Quellen von FORTech, Bernd Paysan, Claus Vogt, Ulrich Hoffmann, Arndt Klingelberg etc...

Dienstleistungen und Produkte von Forthlern und/oder für Forthler (Anzeige)

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 bis 60 Computer, Forth+Assembler PC / 8031 / 80C166 / RTX2000 / Z80 ... für extreme Einsatzbedingungen in Walzwerken, KKW, Medizin, Verkehr / >20 Jahre Erfahrung.

Ing.Büro Klaus Kohl

Tel.: 0+8233-30 524 (Fax: --9971)
Postfach 11 73
D-86406 Mering

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

FORTech Software GmbH

Tel.: 0+381 -405 94 71 (Fax: -4059.471)
Joachim-Jungius-Str. 9
D-18059 Rostock

PC-basierende Forth-Entwicklungswerkzeuge, System comFORTH für DOS und Windows, Cross- und DownCompiler für diverse Microcontroller, Controllerboards mit 80C196, 80C537 und H8, Softwareentwicklung für Microcontroller und PC's, auch unter Windows (und fremdsprachig)

Dipl.-Ing. Arndt Klingelberg

Tel.: 0+2404 -61648 (Fax: -63039)
Strassburgerstr. 12
D-52477 Alsdorf (b. Aachen)

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

ETA Elektrotechnische Apparate GmbH

Tel.: 0+9187 -10.0 (Fax: -10.397)
Industriestr. 2-8
D-90518 Altdorf (b. Nürnberg)

Produkte für Echtzeitanwendungen
FRP1600: Echtzeitprocessor optimiert für Forth
RP-PB1: FRP1600 Prototyping Board.

Möchten auch Sie oder Ihre Firma hier aufgeführt werden? Bitte wenden Sie sich an die Anzeigenverwaltung (s. Impressum).

Ihre Anzeige plus 3 Zeilen je 45 Zeichen Text kosten 90.-DM (incl. 20.-DM Einrichtung/Änderung, je Zusatzzeile 10.-DM) und das komplett für ein ganzes Jahr.

Dienstleistungen und Produkte von Forthlern und/oder für Forthler (Anzeige)

IMPRESSUM

Name der Zeitschrift

FORTHMAGAZIN - VIERTE DIMENSION
Organ der Forth-Gesellschaft e. V.

Herausgeberin

FORTH-Gesellschaft e. V.
Postfach 1110
85701 Unterschleißheim
Tel./Fax: 089/3173784
Mail: secretary@admin.FORTH-eV.de

Redaktion & Layout

Claus Vogt
Ebersstr. 10
D-10827 Berlin
Tel.: 030/782 81 79 (Fax & BBS nach Bedarf)
Mail: vd@FORTH-ev.de

Anzeigenverwaltung: Ulrike Schnitter c/o Forth-Ges.; PF 1110; 85701 Unterschleißheim.

ANS-Forth: Ulrich Hoffmann; uho@informatik.uni-kiel.de; Sehestädter Str. 26; 24340 Eckernförde.

Forthinternational: Fred Behringer; Planegger Str. 24; 8124 München.

Zeichnungen: Rolf Kretzschmar;

Titelbild: Hans-Georg Schmid

Redaktionsschluß '96

Erste Januar-, April-, Juli-, und Oktoberwoche.

Erscheinungsweise

Viermal im Jahr.

Preis

Einzelpreis: DM 10,-

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 beliebige Medien ist auszugswise nur mit genauer Quellenangabe erlaubt. Die eingereichten Beiträge müssen frei von Ansprüchen Dritter sein. Veröffentlichte Programme gehen - soweit nicht anders vermerkt - in die Public Domain über. Für Fehler im Text, in Schaltbildern, Aufbauskizzen etc., die zum Nichtfunktionieren oder evtl. 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.



"Technik - Musik - Kunst - Gestaltung"

Unter diesem Titel wird das Forth-Magazin am Rande der Forth-Tagung'96 (s.u.) eine kleine Ausstellung durchführen, um mit dem Cocktailglas in der Hand über die künftige Gestaltung des Forth-Magazins zu reden und über die technischen Probleme, die der Kunst dabei im Weg stehen. Frank Rothkamms Musik wird auch erklingen - wenn Forth-Algorithmen die Charts erklimmen, soll man sie auch hören.

Auch ohne Kunst hat die Technik so ihre Tücken. Der größte bisher entdeckte Fehler im Weihnachtspäckchen: HOLON darf nicht direkt von der CD 'forthige Weihnacht 95' gestartet werden. Stattdessen sollte ARCHIVE\HOLON.ZIP auf eine Festplatte mit genug Platz expandiert werden.



Die letzten Weihnachtspäckchen sind noch zu vergeben (s. linke Seite). Da ausländische Forth-Mitglieder praktisch keine Chance zum pünktlichen Bestellen hatten, werden sie erstmal bevorzugt. Aber auch die Deutschen (ich habe nichts gegen Deutsche, einige meiner besten Freunde sind Deutsche) kriegen zumindest die CD, die auch weiter bei Klaus Kohl (Adresse s. hintere Umschlagseite) zu einem ähnlichen Preis bezogen werden kann.

Wenn diese Zeilen lesbar sind und auch das Titelbild seiner künstlerischen Qualität nicht völlig verlustig gegangen ist, dann hat die Technik des Forth-Magazins in unermüdlicher Arbeit einen weiteren Quantensprung vollzogen. Die vorliegende Ausgabe ist voraussichtlich komplett ohne Papiervorlage direkt von Diskette gedruckt. Der Umstieg auf Postscript ist - dank des neuen Luxusdruckers - vollzogen. Hinweise auf postscripttypische Fehler wie verschluckte Grafikteile und Textreste, durch Courier ersetzte Zeichensätze etc. sind hochwillkommen. Denn auch die Redaktion hat den Königsweg zwischen professioneller Investition und hobbyistischer Blitzkrücke noch nicht gefunden...



Claus Vogt, Editor des Forthmagazins 'Vierte Dimension'

Einladung zur Forth Jahrestagung '96

vom 19. bis 21. April 1996 in Höfchen (bei Mittweida)

Hotel Höfchen; Moritzfelder Str. 1a D-09648 Höfchen (s.o.)

Organisation:

Thomas Beierlein
Thomas-Mann-Str. 9
D-09648 Mittweida
Tel. (03727) 90941

sofort anmelden!

Am Sonntag findet die Mitgliederversammlung der FORTH-Gesellschaft e. V. statt.

Mitgliederversammlung

Einladung zur Mitgliederversammlung 96 der Forth Gesellschaft e. V.

Hiermit möchte das Direktorium alle Mitglieder der Forth Gesellschaft zur Mitgliederversammlung am 21.04.1996 in Höfchen bei Mittweida einladen. Der Tagungsort ist das Hotel Höfchen in der Moritzfelderstraße 19.

Die Mitgliederversammlung soll sich mit folgenden Tagungsordnungspunkten befassen:

1. Rechenschaftsbericht des Direktoriums
2. Kassenbericht und Bilanz 1995
3. Aussprache, Entlastung, Wahl des Direktoriums
4. Satzungsänderungen

4.1 Änderung des Wahlrechts für fördernde Mitglieder:

Der letzte Satz von Punkt 3 der Satzung soll sich wie folgt ändern:
Natürliche und juristische Personen können "Fördernde Mitglieder" werden. Für sie gelten die Bestimmungen der Satzung entsprechend. Juristische Personen benennen eine natürliche Person, welche die Mitgliedsrechte wahrnimmt.

4.2 Satzungsänderung im Auftrag der Finanzbehörde entsprechend Mustersatzung für gemeinnützige Vereine in Bezug auf Punkt 16 "Auflösung":
Bei Auflösung des Vereins oder bei Wegfall steuerbegünstigter Zwecke fällt das Vermögen des Vereins an eine juristische Person des öffentlichen Rechts oder eine andere steuerbegünstigte Körperschaft zwecks Verwendung für Wissenschaft und Forschung.

5. Arbeit und neue Projekte in der Gesellschaft
6. Verschiedenes

Ergänzende Tagungsordnungspunkte sind bis zum **1. Februar (eventuell entsprechend Erscheinungsdatum der VD ändern)** über das Forth-Büro dem Direktorium einzureichen.

Birgit Steffenhagen

Inserenten

- 2 Forthworks
- 2 cpFORTH
- 2 Forth-Magazin CD
- 36 FORTech

Rubriken

- 3 Editorial
- 3 Impressum
- 3 Forth Jahrestagung'96
- 6 Leserforum
- 6 ff Korrekturen
- 35 Adressen & Ansprechpartner

Forth-Gesellschaft

- 7 Arbeitsgruppe Marc4
- gemächliche Fortschritte
- 7 Forthgruppe München
- 7 Direktorium
- 8 Forth-Magazin
....Korrekteure gesucht
....Technik - Musik - Kultur - Gestaltung
....neues Cover?
....statistische Ergänzungen
- 9Hinweise für AutorInnen

Termine

- 9 Rochester Conference '96

Forth-Systeme

- 9 win32forth 3.0 wieder unter win32s
- 10 F-PC 3.6 mit äöü
- 10 ZF mit Präfixen
- 10 PASM.HLP kann Assembler!

Prozessorgeflüster

- 8 ⚡ FRP 1600

Zeitschriften

- 4 ELRAD-CD
- 10 Elektronik 3/96: Graycode
- 10 Elrad 2/96: CRC
- 10 Forth Dimensions: AutorInnen gesucht

Forth Online

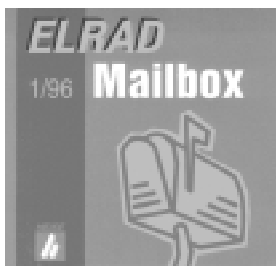
- 10 Taygeta goes Europe ...

Was noch

- 7 Telekom:Doch noch gute Nachricht?
- 8 fLux: George Lucas

Was fehlt:

Die Rubriken Autorengeschenke, ANS, Forth inside, Bücher, Anfänger, Metasprachen und Produktinfos, Conrad Zuse (⚡)



Die ELRAD-Mailbox-CD 1/96 - jetzt auch in der KBBS, der Mailbox der Forth-

Die ELRAD hat sich jetzt gerächt und auch eine CD herausgebracht. Außer dem Abzug Ihrer Mailbox bringt sie die Register-Updates auch von c't und iX, Listings, Software, Service. Softwareproben, Treiber Infodateien rund ums Elektrische. Das neue eMedia-Navigationssystem ist wirklich schnuckelig. Zusammen mit FILES.BBS, HTML-Infodateien und einem eigenen Format (das auch geschrieben werden kann), gestaltet sich die Übersicht


und der Installationsvorgang sehr einfach. /clv.



Der größte bisher entdeckte Fehler im Weihnachtspäckchen: HOLON darf nicht direkt von der CD 'forthige Weihnacht 95' gestartet werden. Stattdessen sollte ARCHIVE/HOLON.ZIP auf eine Festplatte mit genug Platz expandiert werden.

Vielen Dank an Wolf Wejgaard für diesen Hinweis

Portables Forth: Atari-Portfolio  von Wolfgang Führer 12
 Klein genug, um es überall mit hinzunehmen.

cpForth für OS/2  von P. Maier-Schuler 13
 Informationen direkt vom Hersteller.

Forth International von Fred Behringer 16
 Neues aus Schweden und Frankreich
 'Gehaltvolles' aus der 'Forth Dimensions' von Sept/Okt'95


Jetzt geht's ANS FORTH! von Ulrich Hoffmann 17
 Der ANS-Standard hinterläßt Spuren auf CDs, in Netzbrettern
 und in der deutschen Sprache...

Forth auf eine professionelle Basis heben von Peter Knaggs..... 19
 Wie lassen sich Renommé und Verbreitungsgrad
 der Programmiersprache Forth insbesondere bei kommerziellen Nutzern
 verbessern? Hier könnte ein besonderer 'Fachverband' helfen.


Forth braucht keinen qualitätsüberwachenden Fachverband von Fred Behringer 21
 Fred Behringer macht da ganz andere Vorschläge.
 Für den überzeugten Hobbyisten zählt seine eingene Liebe zur Sprache,
 selbst wenn alle Welt sie zur 'Nichtsprache' erklärt.


Jeff Raskins "Canon Cat" von Rafael Deliano..... 25
 Daß Jeff Raskin den Macintosh entwickelte, ist bekannt. Auch daß es
 Forth-Computer gibt. Aber deshalb gleich mit Kanonen nach Katzen schießen?
 Und was hat das mit dem **Toaster** zu tun?

Forth Online von Olaf Stoyke 28
 Eine neue Kolumne über Forth im Internet, WWW usw.

Noch Platz !?!  von Arndt Klingelberg .30
 Ein fast schon historischer Artikel über die Fest-(tags-)Platte, den 16-bit Zahlenraum
 oder den Anfänger-Kopf

Das müssen wir analysieren ... von Claus Vogt 33
 Auch die Redaktion hat ihre liebe Not mit dem Titelthema

Rätsel  von Martin Bitter 34
 Ein wirklich pfiffiger Algorithmus. Aber was macht er?

 = Listing auf Diskette

Nur noch ganz kurz..

kann man die Forth-Weihnachtspäckchen bestellen. Die CD 'forthige
 Weihnacht 95' enthält die Listings der VD des letzten Jahrganges und aller
 vorhergehenden Jahrgänge, die auffindbar waren. Nebenbei auch noch
 zwei komplette Ftp-Server und Quelltextsammlungen von Ulrich
 Hoffmann, Claus Vogt und anderen fleißigen Forthlern.

1	2.387	24.03.95	6:50	vdl961.lst
2	1.834	4.01.96	21:26	cpforth\list1.lst
3	1.448	4.01.96	21:28	cpforth\list2.lst
4	1.007	4.01.96	21:29	cpforth\list3.lst
5	237	19.01.96	2:26	cross\gebuehr.dat
6	1.525	18.01.96	14:32	cross\tarife.dat
7	9.105	9.01.96	4:03	multisk\clock
8	4.513	6.05.95	19:32	new_numb\new_numb.seq
9	180.241	27.02.95	20:09	pasmhlp\pasm.hlp
10	5.342	13.01.94	0:47	platz\dSPACE.seq
11	1.538	16.01.96	19:21	raetsel\raetsel.lst

Gleich bestellen! Info auf Seite 2 Mitte



Reaktionen auf die letzte VD

Just in Time

Ein großes Lob für die termingerechte Fertigstellung der VD!!!

Bernd Beuster, Dez '96

Wie bleibe ich Mitglied?

Der letzten VD lag eine Zahlkarte bei (seufz), schon wieder ein Jahr rum. Aber wie hoch war doch gleich der Beitrag? Nach längerem Suchen wurde ich dann in Heft 4/95 fündig. Ich denke, wenn wir neue Mitglieder werben wollen, sollte die Werbung in eigener Sache etwas augenfälliger erfolgen, am besten mit einer Anmeldung zum Ankreuzen und Ausschneiden. Und wo gibt es schon einen Usenet-Zugang für nur 5,33 DM monatlich (Auszubildende und Studenten) und eine tolle Zeitschrift gratis?

Wieviele MB darf ich eigentlich 'kostenlos' posten?

Gruß Helfried, HSc@arson.forth-ev.de (Helfried Schuerer), Dez '96

[Die Mitgliedschaft kostet 176.- für Firmen/Förderer, 96.-DM für ordentliche und 64.-DM für nachgewiesenerweise arme Mitglieder.

Eine arme, aber ordentliche Firma zahlt wohl am besten per Einzugsermächtigung: Tel/Fax: 089 / 317 37 84]

Klinken ...

Wenn man näher hinsieht, sind viele Kinken drin - dadurch entsteht der Eindruck von "auf die Schnelle". Es liegt mir der anglophile Teil der VD im Magen - ich habe eine Frage in de.c.l.f., ob das egal ist oder nicht.

Jörg sagte mir, die VD wäre Dir zu viel Arbeit, in deinem Artikel sprichst Du von 3 Std. in einer 7 Tage Woche. Ich habe im Moment das Verständnisproblem, daß ich mir einbilde, daß ein WpWin im hui so eine Zeitung layoutet (wie ich uHo mal kurz zeigte) und ein WpWin einige Probleme nicht hat: Das Einlesen von ASCII-Listings geht klick-klick.

Wo liegen die Schwierigkeiten nun genau?

Ciao, Joerg.staben @jrgforth.forth-ev.de

Antwort: Im Artikel steht: 'Meine Arbeitszeit für die VD beträgt derzeit etwa 20 Stunden pro Woche'. Weiter unten steht, daß ich den Aufwand, der durch häufig nötige Umstellung der Software und Technik entsteht, gerne

reduzieren möchte. Dort steht auch, daß ich beim Layout vor allem das zeitaufwendige Layout der Kurzmeldungen automatisieren möchte.

Genau liegen die Schwierigkeiten so: Ich lese die Meldungen derzeit mit WinWord ('hui') ein und verbringe pro Stück vielleicht eine Viertelstunde damit, den Autor aus dem Header (meist Mails) unter die Meldung zu verschieben. Dann werden Header und Footer gelöscht, Umlaute und Schreibfehler korrigiert und die Absatzformate eingestellt.

Wenn Einrückungen, eingestreute Listings oder andere Spezialitäten drin sind, wird es natürlich komplizierter. Zuletzt werden die (in Mails sehr häufigen) Zitate entfernt, gekürzt oder in Hochkommata eingeschlossen, bevor alle Zeilenwechsel im Fließtext entfernt werden. Eine genauere Beschreibung enthalten die Wettbewerbsunterlagen, die Teil des Weihnachtspäckchen sind.

Mailbuch

"Mailbuch" ist eine nützliche neue Rubrik. Die clf-mails waren lesenswert. 80% der EMAIL ist üblicherweise trash. Das macht EMAIL-Lesen zu einer solchen Zeitverschwendung. Wenn man die Mails mit relevanter Information aussortiert und auf 3 Seiten in der VD bringt, ist vielen geholfen.

Für den lesenswertesten Artikel des Hefts halte ich "Parallel-Forth" von Montvelishsky. Weniger wegen des FORTH-Codes, sondern wegen der angesprochenen Grundlagen und der Gegenüberstellung von OCCAM und Linda.

Zu den "Highlights 95" würde ich zählen, daß die VD über den Vereinstellerrand hinausgeht, Material aus clf und dank Behringer aus dem übrigen Europa bringt.

Rafael Deliano, Jan '96

zum Titelbild 'Weihnachtsmann' von Hans Schmid (VD 4/95)

Diesmal war zwar ein Bild vorhanden, aber über den Humor kann man streiten. Außerdem zuviele Stichworte in zu kleiner Schriftgröße.

Rafael Deliano, Jan '96

Das Titelbild ist ja schon fast obszön (aber schön). Sogar meine forth-begeisterte Frau (prust) hat ein bißchen gelächelt.

joerg.plewe@jpsforth.forth-ev.de (Joerg Plewe), Dez '95

Leserforum



zu 'Ein Forth-Programmierer ...' in VD 4/95, Seite 23

Die VD gefällt mir übrigens gut, am besten fand ich Deine Kolumne über die 1, 2, 3, ... Forth-Programmierer.

woi@baltic.e-technik.uni-rostock.de (Egmont Woitzel), Dez '96

... hab gerade die VD bekommen. Die Drohung mit dem Entzug der WeihnachtsCD habe ich bitterernst genommen und Dir mit SnailMail heute einen V-Scheck abgeschickt. Ich möchte die CD ob mit oder ohne Weihnachtspaket, ist mir egal.

An der Glosse(?) Ein Forth-Programmierer... scheint mir ein Körnchen Wahrheit drin zu liegen. Gefällt mir.

Tschüß, Wolfgang All@business.forth-ev.de (Wolfgang Allinger), Dez '96

zu 'VD - quo vadis' von Claus Vogt in VD 4/95

Zeitungen werden gerastert. D.h. es gibt eine Spaltenbreite und feste Längenschritte. Der Redakteur quetscht das Material in dieses Format. Entweder er kürzt den Text oder er verlängert ihn. Für den Druck werden die Schnippsel zusammengesetzt bis die Seite voll ist.

Viele Autoren haben ihr eigenes DTP-System und können deshalb ihre Schnippsel fertig layouten. Heute wird bei meinen Artikeln immer

doppelt gemoppelt. Erst layoute ich die Seiten auf dem Amiga. Dann verschicke ich die Teile. Dann layoutest du das ganze nochmal. ...

Wenn man sich grob auf gängige Schrifttypen (Times Roman) und Schriftgröße (10 US-Punkte) festlegt, dürfte das Blatt auch nachher noch homogen aussehen.

Das automatische Erfassen der ASCII-Texte ist jedenfalls kein ausreichender Schritt nach vorn. Druckfertiges Material muß das Ziel sein. Das setzt aber immer voraus, daß dein DTP-System Postscript importieren und previewen kann. Und das tut das Spielzeug eben nicht. Kretschmar und Wilke hatten da deutlich bessere DTP-Systeme.

Wegen mangelnder technischer Voraussetzungen und dem sich daraus ergebenden Frust bei der Erstellung, wird das Ergebnis spätestens nach acht Ausgaben wohl wieder ein ausgebrannter Editor sein. Wie gehabt.

jrd, Jan '96

zu 'Betreff: Forth Magazin sucht Programmierer' in VD 4/95, Seite 24

Die Urkunde auf S.26 war gut gemacht, wurde dann aber zu sehr verkleinert. Machst du keine Probeausdrucke?

Rafael Deliano, Jan '96



Leserbriefe:
Am liebsten kurz. Sonst trifft uns die Pflicht zur Kürzung.
Die Redaktionsadresse lautet:
Forth-Magazin 'Vierte Dimension'
Claus Vogt, Ebersstr. 10,
D-10827 Berlin, vd@FORTH-ev.de

Forth. Kurz und Knapp. Das

Habe das Problem in VD 4/95 auf S.24 nicht mitbekommen. Hier fehlte sozusagen die 'Quote' im Artikel. Statt '60 Zeilen Problembeschreibung' hätte ich wenigstens eine auszugsweise Beschreibung wie ' ' in umwandeln o.ä. erwartet. Die folgenden News waren schön zu lesen, aber worum gings denn? Naja, schön daß es Dir die Arbeit erleichtert hat...

Gruß Helfried
HSc@arson.forth-ev.de (Helfried Schuerer)

[Die 60 Zeilen mit dem Eingabeformat des DTP-Systems hätte ich wohl besser dringelassen. Die Originalmail findest Du z.B. auf der CD im File USENET/DCLF]

nochmal 'kleinstes Forth'

Rafael Delianos Antwort in VD 4/95, S.6 auf meine Anfrage in VD 3/95, S.6 ist interessant und aufschlußreich. Gemeint war aber meine Frage im mehr theoretischen Sinn: Wieviele Worte sind höchstens nötig, um ein Forth-System aufzubauen, das dann aus sich selbst heraus gestattet, andere Worte hinzucompilieren? Zwei Beispiele aus anderen Gebieten:

- Mit wievielen verschiedenen Farben läßt sich eine Landkarte so einfärben, daß keine zwei aneinandergrenzende Länder dieselbe Farbe haben? Die Antwort lautet: Vier. Fünf sind nicht nötig, mit drei geht es nicht. Lange Zeit war das nur eine Vermutung. Seit einiger Zeit hat man einen Beweis.

- Mit wievielen (zweistelligen) Operatoren läßt sich die Aussagenlogik aufbauen? Mit einem. Wenn ich mich nicht irre, ist das der Pierce-Operator, den wir Datenblatt-Leser, die wir mit der Shannonschen Schaltalgebra vertraut sind, NAND nennen.

Zurück zu Forth: Man müßte natürlich

(1) genau definieren, was man als "Wort" zulassen möchte (ist NEXT ein solches?),

(2) genau sagen, was man unter einem eigenständigen Forth-System verstehen will (den ganzen DOS-Sockel natürlich nicht mitrechnen?).

Meine Fragerei zielt selbstverständlich darauf ab, zu erfahren, wie klein ein Metacompiler gehalten werden kann. Wieviel Forth muß sein, damit es sich selbst "aus dem Sumpf ziehen kann"?

Fred Behringer, München

**Telekom:
Doch noch gute
Nachricht?**

Hier meine persönliche Interpretation der neuen 'Gebührenvielfalt': *Gebührenviel* für die Telekom und für uns *falt* :-).

Ortsverbindungen zwischen 18 und 8 Uhr werden um **161 bzw. 57% teurer**. Schlecht für Online-Chats :-).

Fernverbindungen zwischen 18 und 8 Uhr werden um **11% teurer**, zwischen 21 und 5 Uhr jedoch **12% billiger**, werktags zwischen 2 und 5 Uhr sogar um **82% billiger**. Super für pollende Life-Sites mit Zeitschaltuhr :-).

Ab 1996 kostet eine Fernverbindung im günstigsten Fall (werktags zwischen 2 und 5 Uhr) nur noch das Doppelte einer Ortsverbindung, gegenüber bisher dem 17fachen. Das ist jetzt sogar mit 2400 Baud Modem günstig.

Im **Gebührenvergleich** im Bild habe ich, in Anlehnung an das neue Tarifsystem, die für die Datenfern-

übertragung interessanten fünf Tarifzeiten dargestellt. Die Kosten von 1995 basieren auf dem Billigtarif, der werktags von 18 bis 8 Uhr und am Wochenende durchgehend galt. Vergleichsbasis ist ein 12 min-Gespräch. **An Wochenenden und Feiertagen entfällt Tarif Fern3** :-), dafür gilt City1 bzw. Fern1 von 5-21 Uhr und City2 bzw. Fern2 von 21-5 Uhr.

HSc@arson.forth-ev.de (Helfried Schürer)

[Wer unter Crosspoint die neuen Gebühren berechnet haben möchte, findet die nötigen Dateien auf der Listingdiskette/red]

**Forthgesellschaft:
Arbeitsgruppe Marc4
gemächliche Fortschritte**

Die OTP-Version M48C260 ist über Enatechnik bestellt, aber es gibt noch keinen festen Liefertermin, weil sie TEMIC noch nicht freigegeben hat. Die ersten Exemplare sind vermutlich noch teuer (40DM), aber vielleicht ändert sich das im Verlauf des Jahres. Der Chip kommt im DIL28 Gehäuse. Das ursprünglich angekündigte SSO28 (das ist schlimmer als SO28) hätte sowohl manuelles Löten als auch Sockeln unmöglich gemacht.

Die erste Ausgabe des Newsletters ist fertig. Sie befaßt sich mit der Architektur der CPU. Erhältlich von Klaus Kohl gegen 5DM in Briefmarken Compiler und Simulator existieren jedoch noch nicht.

jrd, Jan'95

**Forthgesellschaft:
Forthgruppe München**

Jeweils am vierten Mittwoch des Monats trifft sich die Münchner Gruppe beim Chinesen XIAN Morun-

MARC4
Gemeinschaft für die Entwicklung von Forth-Systemen

INHALT

1. Einleitung
2. Die neue CPU-Version
3. Die neue CPU-Version
4. Die neue CPU-Version
5. Die neue CPU-Version
6. Die neue CPU-Version
7. Die neue CPU-Version
8. Die neue CPU-Version
9. Die neue CPU-Version
10. Die neue CPU-Version
11. Die neue CPU-Version
12. Die neue CPU-Version

genstr.8, S-Bahnstation Westkreuz. So geschehen am 24. Januar. Mit 8 Anwesenden etwas besser besucht als üblich.

Ab und zu werden technische Vorträge angekündigt und gehalten. unlängst z.B. über LIFO eine LISP/FORTH Promenadenmischung. Manchmal wird der Inhalt des für das nächste Treffen angekündigten Vortrags von den Anwesenden durch Akklamation als wenig ergiebig eingeschätzt. Der Referent wird dann ermuntert das Thema verstärkt schriftlich aufzuarbeiten und an die VD-Redaktion zu schicken. Da die Veranstaltung im Dezember jedoch ausfällt, hat man sich am Jahresanfang lang nicht gesehen und genügend anderes zu besprechen.



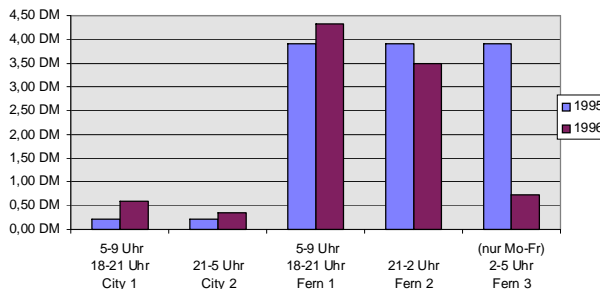
In Sachen FORTH gab es deshalb diesmal nur ein kurzes Scharmützel zum Thema Groß- und Kleinschreibung. Man kommt bei derartigen spontanen Diskussionen meist nicht richtig zum Zug, weil man den Mund voll Reis und Morcheln hat. Jedenfalls nicht ohne Gefahr für die Tischdecke.

Einig waren sich wie üblich alle darüber, daß Bill Gates eine Bedrohung der Menschheit darstellt. Uneinig darüber, wem die gerade frisch gelieferte Frühlingsrolle gehört.

jrd, Jan'96

**Forth-Gesellschaft:
Direktorium**

'Was macht eigentlich das Direktorium?' fragen uns immer wieder die Mitglieder, die ja im April auf der Forth-Tagung wieder zu wählen haben. Allgemein besteht die Vermutung, daß das Direktorium 'NICHTS!' mache. Nun mangelt es der Redaktion zwar an Zeit und Fachkenntnis zur Erforschung quantenphysikalischer Phänomene, aber telefonieren können wir. Unseren Direktor Jörg Plewe erwischte es als erstes. Als kooperati-



Telekom: Was kosten zwölf Minuten?

ver Mensch neigt er überhaupt nicht zum Disput und bestätigte die von uns geäußerte Auffassung, das Direktorium mache tatsächlich im wesentlichen 'NICHTS!'. Als nächstes befragten wir (ebenso unauffällig) Direktorin Birgit Steffenhagen. Ihre Antwort war weit ergebiger. Das Direktorium hätte im Jahre 1994 sehr viel getan, als es der Forth-Gesellschaft nicht so gut ging. Im Jahre '95 sei tatsächlich wenig los gewesen.

Wie dem auch sei. Wir und unsere Leserinnen und Leser wünschen sich natürlich vom alten und neuen Direktorium etwas authentisches, selbstgeschriebenes in der VD lesen zu können.

Claus Vogt

Forth-Gesellschaft: Forth-Magazin

Last-Minute-Korrekturen in Notfällen: Wir suchen Freiwillige mit Faxgerät und Schulkenntnissen in der deutschen Sprache. Ich würde Euch zwei Tage vorm Notfall kurz Bescheid sagen und bei Eintreten desselben ein kurzes Testfax senden. Wenn Ihr dann am Fax sitzt, kommen die Korrekturfahnen, bis Ihr 'Stop' sagt. Wenn nicht, muß es halt weiter wie bisher gehen....

Bei Interesse bitte eine kurze Nachricht an die Redaktion (030/782 81 79). Es tut nicht weh, definitive Zusagen sind nicht nötig!

Claus Vogt

„Technik - Kultur - Musik - Gestaltung“

Unter diesem Titel plant die Redaktion des Forth-Magazins 'Vierte Dimension' am Rande der Forth-Tagung '96 im April eine Vernissage, die die ästhetischen Aspekte der Technikentwicklung beleuchten soll. Mit einem Cocktailglas in der Hand und einem Ambiente von Plastiken, Lyrik, Musik, Videos, Postern und Grafiken sollte hier nicht nur die neue Gestaltung der VD zur Sprache kommen. Die Exponate dürfen im weitesten Sinne mit Forth zu tun haben. MitstreiterInnen sind hochwillkommen - um eine kurze Nachricht an die Redaktion wird gebeten.

neues Cover...

(..) Es liegen zwei Beispiele für Cover bei. Scannen und ähnlich wie beim CreepingFORTH-Artikel als Illustrationen verwenden. Das Blatt der University of Strathclyde "INTERFACE" ist in dieser Hinsicht vorbildlich gemacht.

Man könnte (...) das Blatt dabei umbenennen. Der Titel des Vorbilds "FORTH-Dimension" ist im Englischen ein nettes Wortspiel. In der deutschen Übersetzung "Vierte Dimension" geht der Bonmot jedoch verloren. Da der Inhalt des Blatts FORTH ist, sollte das unmittelbar aus dem Titel hervorgehen. Man kann es zwar "FORTH-Times", "FORTH-Gazette" usw. nennen. Aber der klarste Titel ist eben "FORTH" ohne redundantes Anhängsel.

Wer die Hefte sammelt, ordnet sie chronologisch. Deshalb sollte Heftnummer und Jahr deutlich auf dem Cover vermerkt sein.

Das bisher auf dem Cover gebotene abgemagertes Inhaltsverzeichnis ist

redundant, weil innen ohnehin ein komplettes Verzeichnis vorhanden ist.

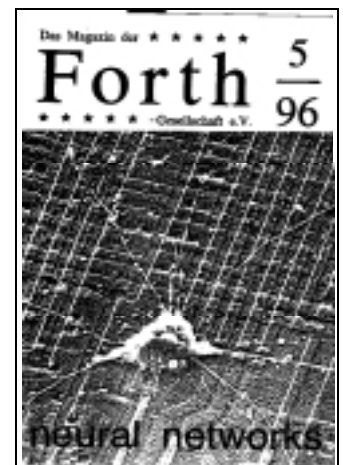
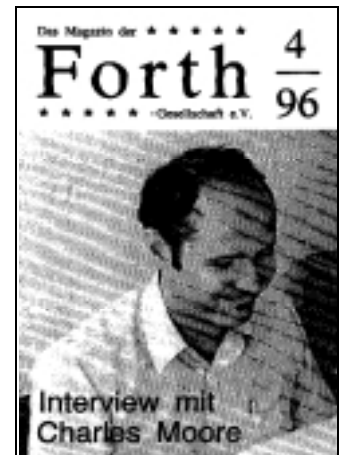
Ein Foto als Cover wirkt wesentlich moderner. Es sollte Bezug zu einem Artikel im Heft haben. Daraus ergibt sich der Text der Titelzeile zum Foto. Bringt man weitere Titel auf dem Cover, die auf Artikel im Heft verweisen, sollten sie wesentlich kleiner und unauffälliger als der Haupttitel sein.

Rafael Deliano

flux

(...) aber ich hab noch einen größeren Klops auf Lager: Letzte Woche habe ich einen Soundtrack zu einer 3D Computeranimation für George Lucas abgeschlossen. Das war eine Wiederverarbeitung der "Krieg der Sterne" Musik.

flux@thing.net (Frank Rothkamm)



neues Titelbild? Zwei Entwürfe von Rafael Deliano

Nach schwerem Leiden
von uns gegangen:



FRP 1600

Wir werden ihm ein liebendes
Andenken bewahren.

R.I.P. Die Beerdigung fand auf Wunsch der
Angehörigen in aller Stille statt.

An der Traueranschrift E-T-A bitten
man von Beileidsbezeigungen abzusehen

Ergänzungen

zu 'VD quo vadis?' in VD 4/95, Seite 20

Ergänzend wollen wir hier ein paar statistische Angaben zum Forth-Magazin hinzufügen:

Im Jahre 1995 haben 24 verschiedene Autoren 36 längere Beiträge gebracht, davon trat einer 'nur' als Übersetzer und zwei als Koautoren in Erscheinung. Drei Künstler machten Zeichnungen. Leider steht keiner von ihnen regelmäßig zur Verfügung. Die grafische Gestaltung der 36 ausgewerteten Beiträge ist insgesamt nicht großartig. Nur 15 enthielten wenigstens ein grafisches Gestaltungselement. Von den anderen 21 Beiträgen enthielten 7 immerhin mindestens eine Tabelle oder einen separaten Textka-

sten, weitere 8 ein Listing. Die anderen 6 Beiträge waren reine 'Bleiwüsten'.

7 der 24 Autoren hielten auch auf der Tagung '95 einen Vortrag. Bei den Tagungsbänden ist die grafische Gestaltung besser. Hier sind neben den typischen Vortragsfolien zahlreiche Schaltpläne und Prinzipskizzen zu bewundern. Für den Tagungsband '95 reichten von den ca. 50 Teilnehmern 16 Beiträge ein, die sich in Hinsicht ihrer grafischen Gestaltung 13:2:0:1 auf die vier Kategorien 'mit grafischem Element': 'mindestens ein Kasten/eine Tabelle': 'mindestens ein Listing': 'reiner Text' verteilen. Im Tagungsband '93 reichten 15 von 61 Teilnehmern Beiträge ein (grafische Gestaltung 8:2:4:1) und im Tagungsband '91 waren es 20 von 60

(9:1:2:1). Die anderen Tagungsbände liegen mir leider nicht vor.

Insgesamt würden wir uns für das Forth-Magazin 1996 wünschen, daß - ähnlich wie auf den Tagungen - etwa ein Drittel der LeserInnen auch in den AutorInnen-Kreis vorstoßen und ihre dann wohl ca. 200 Beiträge zur Hälfte mit Grafiken oder Bildern schmücken.

Hinweise für Autorinnen und Autoren:

Grafiken bereiten manchmal Konvertierungsprobleme. Als Formate hätten wir bei Grafiken gerne gängige Grafikformate wie HPGL, EPS, GIF, TIF und PCX. Eine große (max. A4) reproduzierfähige Papiervorlage läßt sich scannen oder in die Druckvorlage einkleben. Sie kann manche Sorgen ersparen. Falls Sie ein Fax oder noch besser Faxmodem haben: Es kann bei geringerem Qualitätsverlust das Scannen ersetzen. Faxen möglichst A4-füllend und mit hoher Auflösung (ca. 200 dpi). Wenn mein Fax nicht eingeschaltet ist, hinterlassen Sie eine Nachricht, wann Sie es wieder probieren möchten.

Artikel erhalte ich am liebsten per Brief sowohl auf Diskette als auch auf Papier. Als Artikelformat haben sich Ascii, RTF, WinWord (möglichst Version 2.0) und WordPerfect bewährt. Eingebettete Objekte, Tabellen und eingestreute Listings sollten nochmal als separates File vorhanden sein. Es ist nicht möglich, TeX- und Postscript-Texte einzulesen. Eine Zusendung per E-Mail ist möglich. Für Rückfragen wüßte ich am liebsten die E-Mail-Adresse und Telefonnummer des Autors, bzw. der Autorin. Für die Zusendung der Korrekturfahnen brauche ich natürlich die Postadresse.

Bei Kurzmeldungen ist E-Mail der günstigste Weg, bei aktuellen Meldungen tut's aber auch der Anrufbeantworter.

Falls Sie zum ersten Mal einen Artikel einsenden: Erstlingswerke nehmen wir besonders gerne. Wir gehen davon aus, daß eingesandte Artikel zur Veröffentlichung bestimmt sind. Falls Sie zunächst keine Veröffentlichung, sondern nur Meinungsäußerungen oder Verbesserungsvorschläge wünschen, vermerken Sie dies deutlich sichtbar. Auch sonst treten wir an Sie heran, wenn wir stilistische oder inhaltliche Korrekturen für wünschenswert halten.

Claus Vogt, Editor des Forth-Magazins

Korrekturen zu den Weihnachts-Sonderdrucken

In beiden Sonderdrucken befinden sich geheimnisvollerweise doppelte Seitenzahlen. Schuld ist ausnahmsweise nicht das DTP-System, sondern der Kopierer. Inzwischen gibt es Kopierer, die einen ähnlichen IQ wie das DTP-System aufweisen. Die richtigen Seitenzahlen erhalten sie am einfachsten, indem Sie zur Titelseite 'eins' sagen und dann jeder Folgeseite den Successor der letzten Seitenzahl zusprechen (sog. 'durchzählen').

Vielen Dank für Ihr Verständnis,
Claus Vogt

Korrekturen zum Weihnachts-Sonderdruck "Moerser Spiel"

Autor des einleitenden Artikel ist Michael Major (Moers) und nicht - wie fälschlicherweise angegeben - Friedrich Prinz. Wir bitten um Entschuldigung

clv

Leider haben sich in den Kommentaren zu Martins Version des Spiels einige Fehler eingeschlichen. (Fritz: "Das könnte doch in die VD, Aber Ihr müßt Euch beeilen." --> Das gab Hektik und so kam es, daß ein paar Kommentare nur noch aus historischen Gründen im Quellfile sind. Sie können ersatzlos gestrichen werden.

In Zeile 315 sind die Werte für die Tasten falsch angegeben. Richtig muß es heißen:

```
\hoch=0, rechts=1, runter=2, links =3, ESC=4
```

In Zeile 376 muß ab >Deshalb< alles gelöscht werden, ebenfalls die gesamte Zeile 377.

Ich hoffe, daß Eure Irritationen sich in Grenzen hielten.

Martin Bitter, Hehrhoog

'Bechstaben- verwuchselungen' in VD 4/95 ..

Zunächst möchte ich mich bei allen Lesern bedanken, die der Redaktion in schonungsloser Offenheit über die Tippfehler und Satzverstümmelungen in VD 4/95 berichtet haben. Nachdem ich die Ausgabe nochmal überflogen hatte, ging es mir gar nicht gut. Notfälle, wie vor Weihnachten wird es sicher immer wieder geben. Trotzdem ziehen wir folgende Konsequenzen für unsere Arbeitsstruktur und richten eine Bitte an Fax-Eigner:

(a) Ab sofort wird der Redaktionsschluß vorverlegt. Er steht jetzt wieder im Impressum.

(b) Ab sofort wird um Einhaltung des Redaktionsschlusses gebeten. Wem meine ersten Gehversuche ("wird in einer der nächsten Ausgaben...") zu unbeholfen klingen, den bitte ich um bessere Vorschläge. Ich persönlich hätte auch lieber alles gleich erledigt, aber es geht nunmal nicht.

(c) Ab sofort wird eine zusätzliche Korrekturschleife gefahren. Alle Manuskripte gehen sofort nach Eingang in die Korrektur (vielen Dank an Birgit für die Mehrarbeit). Für die AutorInnen heißt das, daß sie bei ernstzunehmenden Korrekturen manchmal auf zwei verschiedenen Wegen informiert werden.

(d) Ab sofort werden Freiwillige mit Faxgerät für Last-Minute-Korrekturen gesucht. Bedingung: Faxgerät, Schulkenntnisse der deutschen Sprache. Bei Interesse bitte eine kurze Nachricht an die Redaktion. Es tut nicht weh - definitive Zusagen sind nicht nötig! Ich würde dann zwei Tage vorm Notfall Bescheid sagen und bei Eintreten desselben ein kurzes Testfax senden. Wenn Ihr dann am Fax sitzt, kommen die Korrekturfahnen, bis Ihr 'Stop' sagt. Wenn nicht, muß es halt weiter wie bisher gehen...

Claus Vogt, Editor

Termine: 1996 Rochester Conference in Toronto

16th Annual 1996 Rochester Forth Conference on Open Systems

June 19-22, 1996

Ryerson Polytechnic University Toronto, Ontario, Canada

Preliminary Call for Papers

The Rochester Conference will again provide a forum for researchers, developers, and vendors to present the latest practical results dealing with open systems. The conference seeks original papers relevant to the design, development, implementation, and use of open systems. Conference topics include:

- The Open Boot Standard - Plug and Play Systems
- Scripting Languages - SGML and HTML
- Distributed Computing - Educational Issues

Other areas of interest are Forth programming standards, embedded systems, real-time systems, and the use of Forth for scientific and engineering applications.

Additional information will be posted on the world-wide web as it

becomes available: <http://maccs.dcss.mcmaster.ca/~ns/96roch.html>

Kontakt:
Nicholas Soltseff
Dept. of Computer Science &
Systems
McMaster University
Hamilton, Ontario
Canada
L8S 4K1
ns@maccs.dcss.mcmaster.ca

Forth-Systeme: win32forth 3.0 wieder unter win32s

STARTUP.TXT Getting Started with Win32Forth by Tom Zimmer (...)

Win32Forth is a public domain ANS compatible Forth language application development system. It was initially written by Andrew McKewan, Tom Zimmer, Robert Smith and Jim Schnieder during 1994 and 1995.

(... History...) Win32Forth took a side road into the area of using only absolute addresses during 1995, but many problems with compatibility with Windows95 and Win32s led me to return to relative addressing as a much more portable solution.

(...) In January 1996, Win32Forth had grown to over 5000 words and 400k bytes of dictionary, which seemed like a lot of overhead to add to a small application program. The solution Andrew came up with was to split the forth dictionary architecture into two parts, application and system. The Application dictionary holds "bodies" of all the words that an application might need to use at program runtime. The System dictionary holds all the non-application words, like those used only during program compile time.... Using this split or dual dictionary architecture reduces a TURKEYED applications overhead by over 200k bytes.

(...)

Enough Already (I Hope)

Well thats about all I have to cover, and I hope it is enough to get you started. There is a lot to learn in Win32Forth, and I believe there are enough tools provided to make it possible to learn alot about Windows programming. If you need more, you can always purchase a "real" development system.

Remember, DON'T PANIC

ZForth@ix.netcom.com (Tom Zimmer), feb'96

Forth-Systeme: F-PC 3.6 mit äöü

Wer KEYB.SEQ benutzt und die deutschen Umlaute und sonstige grafischen Zeichen auch beim Umstieg von F-PC 3.56 auf F-PC 3.6 mitnehmen möchte, sollte nur noch das neue KEYB.SEQ benutzen. Das alte crasht, wenn es mit dem neuen UNHEAD.SEQ (headerlose Worte von Tom Zimmer) kollidiert.

Das neue KEYB.SEQ läuft mit alter und neuer Version. Zu finden ist es in der KBBS in der Datei F356CLV1.ZIP. Dieses Archiv enthält außerdem einige andere allgemein verwendbare Quellen, die mir beim Update von F-PC 3.56 auf 3.6 zu ordnen wert schienen. Das meiste ist bereits auf der CD.

Claus Vogt, Jan '95

Forth-Systeme: ZF mit Präfixen

Ein neues NUMBER ermöglicht in ZF die Eingabe von Präfixzahlen wie z.B. \$FF, %1001. Oftmals sind Quelltexte einfacher zu verfassen und zu lesen, wenn Zahlen in verschiedenen Basisformaten eingegeben werden können. Welche Bits bei einem Wert von 42 gesetzt sind, ist nicht unbedingt offensichtlich. Bei der binären Zahl 101010 ist das schon viel deutlicher.

Von der ATARI-Seite volksFORTHverwöhnt, habe ich die Möglichkeit, Zahlenformate ohne Verwendung von HEX, DECIMAL, BIN etc. zu verändern, sehr vermisst. Zumal HEX in ZF nicht statesmart ist und ein vergessenes DECIMAL zu verwirrenden Ergebnissen im Code führen kann.

Das Listing (NEW_NUMB.SEQ) befindet sich auf der Listingdiskette und auf der Weihnachts-CD. Ich hoffe, die Kommentare im Quelltext sind aussagekräftig genug, um meine Lösung nachzuvollziehen. Bei mir läuft NEU_NUMBER von Beginn an stabil und zuverlässig.

Martin Bitter (Mehrkoog)

Forth-Systeme: PASM.HLP P kann Assembler !

Die wesentlich erweiterte und mit einem deutlich schlechteren Nutzerinterface versehene Datei PASM.HLP dient dazu, das originale PASM.HLP zu ersetzen. Durch das einfache hochfunktionelle Konzept der



F-PC-online Hilfe kann wie gewohnt im F-PC mit H oder ALT-H/F9 Hilfe angefordert werden.

Über die Hilfe zum Prefix-Assembler PASM.HLP hinaus bietet die neue PASM.HLP vor allem Hilfe zu den einzelnen Befehlen des 80x86. Durch die Größe der Datei, die jenseits der 64K-Grenze des SEDitors liegt, kann man zwar im F-PC nicht vollständig durch die Hilfe blättern, einzelne Punkte werden aber korrekt ange-sprungen.

Die detaillierten Informationen zu den 80x86 Instruktionen haben sich bis dato bewährt, weitere Ergänzungen können durch die Forth-Gemeinde gerne vorgenommen werden - ich denke da vor allem an den 486-Assembler des F-95 bzw. Win-32For.

Die Datei PASM.HLP befindet sich auf der Listingdiskette.

Jörg (Jörg Staben), Jan '96

Pressescha u Elektronik: Graycode

Jaromir Kolouch schreibt in der Elektronik, Heft 3, 1996 auf Seite 96-102 über "Gray-Code: Viele Vorteile für PLDs". Bei der Entwicklung von digitalen Schaltungen wird oft der Graycode eingesetzt. Beim Hoch- und Runterzählen im Graycode, z.B. beim Längenmessen mittels Strichcode, jeweils nur ein Bit ändert. Beim Binär-code würden zum Beispiel beim Übergang von 7 auf 8 Millimeter drei Bits gleichzeitig 'springen'.

Werden die Zustände eines Automaten mit Hilfe des Graycode kodiert, kann man die Zustandsänderungen so kodieren, daß sie sich nur in einem Bit unterscheiden. Das vereinfacht die nachfolgende kombinatorische Logik, da hierdurch sogenannte Glitches vermieden werden, das sind kurzzeitige impulsförmige Signaländerungen beim Übergang von einem Zustand in den Folgezustand des Automaten. Würden solche Signale an den Eingang eines Zählers gelangen, wären Fehlschaltungen die Folge.

Ich habe den Artikel zum Anlaß genommen, einen kleinen Binär-Graycode-Konverter in Forth zu schreiben.

FORTH DEFINITIONS DECIMAL

```
: >GRAYCODE ( u1 -- u2 )  
  DUP 1 RSHIFT XOR ;
```

```
: GRAYCODE> ( u1 -- u2 ) DUP  
  31 0 DO  
  1 RSHIFT TUCK XOR SWAP  
  LOOP DROP ;
```

Während die Umwandlung von einer binären Zahl in den Graycode (>GRAYCODE) in einem Schritt vollzogen werden kann, ist dies leider bei der Rückkonvertierung (GRAYCODE>) nicht der Fall.

Als Forthsystem wurde ein 32 Bit ANS-Forth zugrunde gelegt. Wird nur eine Konvertierung von 8-Bit bzw. 16-Bit Zahlen gefordert, braucht die Schleife in GRAYCODE> nur 7 bzw. 15 mal durchlaufen werden. In diesem Fall schadet es auch nichts, wenn die Schleife häufiger durchlaufen wird, da dann immer mit Null geXORed wird, wobei der erste Operand dann unverändert bleibt.

Bernd Beuster, Jan '96

Presseschau Elrad: CRC in ELRAD

In Heft 2/96 ist wieder etwas 4thiges zu lesen: R. Delianos Artikel über CRC in Software mit Source in nanoFORTH. Beschrieben wird die Realisierung mit Schleifen und Tabellen. Ferner die Unterschiede für MSB-first und LSB-first bei den Daten. Schwerpunkt liegt dabei auf 16-Bit-CRCs.

Die VD taucht im Literaturverzeichnis auf, weil Claus Kühnel hier schon mal einen Artikel über CRCs gebracht hat.

Jrd/ea, Januar '96

Forth Dimensions: AutorInnen gesucht

In its 17th year of publication, Forth Dimensions magazine (published in the USA) serves the world's largest Forth audience.

BUT: Forth Dimensions relies on the written contributions of Forth users in order to maintain its high level of service to our readers.

AND: Forth Dimensions needs new articles now!

Subject matter to consider includes:
* tutorials with Forth code that are targeted at beginners,

* stories about interesting applications and how Forth was used to meet particular challenges or performance criteria,

* classic or groundbreaking techniques in Forth,

* discussion or innovations geared to Forth experts,

* education about technical aspects of ANS Forth,

* your improvements/enhancements to material that has appeared previously in our pages,

* letters to the editor,
* and any other Forth-related material that interests you.

This is a great way to help others to benefit from your experience, to solicit feedback, and to make a tangible contribution to the international Forth literature.

Please send e-mail to inquire about your article idea or to ask the best way to submit your work. Articles must be submitted in the English language. We will look forward to hearing from you!

With thanks in advance,
Marlin Ouverson,
Editor Forth Dimensions
ouversonn@aol.com, Jan '96

Forth online: Taygeta goes Europe ...

You may be interested to know that I am now providing a European mirror of the Taygeta WWW area on my own Forth-WWW area, URL below.

Peter Knaggs-
pjk@paisley.ac.uk
<http://www.paisley.ac.uk/~cis/forth/>

Ich bin nicht sicher, ob ich die richtige Seite bei Peter Knaggs erwischte habe oder nicht, aber oberflächlich gesehen sieht das anders aus als <http://taygeta.com/forth.html>. Allerdings verrät der folgende URL schon, daß man unter "Forth Interest Group Home Page" den gespiegelten Bereich betritt: "<http://chaos100.paisley.ac.uk/~cis/mirrors/taygeta/fig.html>". Offenbar sind die Taygeta-Seiten unauffällig in Peters Seiten eingebunden worden. Wie dem auch sei - schneller als das Original ist es allemal.

os@cs.tu-berlin.de (Olaf Stoyke)



F-PC, (kooperatives) Multitasking

von Michael Scholz

Bonhoefferstraße 2; Briefkasten 1340; 69123 Heidelberg

In den Weihnachtsferien kopierte ich mir nach langer Zeit der Forth-Abstinenz das F-PC v3.5614 von Tom Zimmer auf die Platte und spielte ein wenig damit. Um einige Anregungen zu erhalten, holte ich mir ebenso die gesammelten "Vierten Dimensionen" hervor und kramte in diesen. Da ich immer wieder eine Anwendung für Multitasking gesucht habe, brachte mich die Idee aus einer alten VD dazu, die aktuelle Zeit ständig in der oberen rechten Ecke auszugeben. Mit zwei Worten, einer User-Variablen und ohne Assembler-Primitive ist mir dies auch recht schnell gelungen.

Stichworte: F-PC Multitasking

In der Datei `..\SRC\TIMER.SEQ` sind einige Wörter definiert, mit denen man die Ausgabe der Zeit gestalten kann. Das vorhandene Wort `FORM-TIME`, welches `BUILD-TIME` aufruft, schien mir anfangs unbrauchbar, da `BUILD-TIME` die Zeit in der Reihenfolge Stunden:Minuten:Sekunden.Zehntelsekunden (hh:mm:ss.tt) in den Puffer `DTBUF` ablegt. Zehntelsekunden möchte ich aber nicht auf meinem Bildschirm sehen. Daher habe ich die Umwandlung der Zeit mit Hilfe der Wörter aus `..\SRC\TIMER.SEQ` in dem Wort `(.CLOCK)` zum Teil selbst vorgenommen. Das sah so aus:

```

1 : (.clock) ( d --- a ) \ Wandelt die Zeit in die Form hh:mm:ss um.
2   gettime           \ Aus der Datei ..\src\timer.seq
3   base @ >r decimal \ wurde die Umrechnung von Dos-Zeit in
4   dtbuf off         \ ein lesbares Format uebernommen,
5   swap build-hm     \ jedoch nur fuer Stunden, Minuten
6   ." " + split ##+ drop \ und Sekunden.
7   r> base ! dtbuf ;
8 : .clock ( --- ) \ Gibt Stunden, Minuten und Sekunden aus.
9   (.clock) count type ;
10 background: clock [ ... ] ; \ wie unten

```

Diese Version entstand und funktionierte recht schnell. Allerdings war sie noch viel zu kompliziert. Um meinen Ansatz zu vereinfachen, mußte ich mich von kompliziertem Denken sowie von redundantem Code trennen. Ich kann nämlich den von `BUILD-TIME` gefüllten Puffer `DTBUF` doch nutzen, indem ich ihn nur bis zur Stelle der Sekunden ausgabe. Von der Länge, welche `COUNT` liefert, ziehe ich die letzten drei Stellen für Zehntelsekunden ab, und fertig ist der String für die Ausgabe mit `TYPE` in der Form `hh:mm:ss`.

Damit der Background-Task `CLOCK` nicht alle Ressourcen mit der Ausgabe der Zeit verbraucht, habe ich die Idee von T. Zimmer aus seinem Beispielprogramm der Datei `..\SRC\MULTASK.SEQ` übernommen und vor die Ausgabe

eine Entscheidung gesetzt. Ursprünglich war ich der Meinung, daß das Wort `PAUSE` zu Beginn von `CLOCK` ausreichend sei. Das ist ein Irrtum; die Ausgabe ist offensichtlich doch zu ressourcenintensiv. Jetzt wird die Zeit nur ausgegeben, wenn die Dummy-Variable `FOO` durch 1000 teilbar ist. Die restliche Zeit wird mit `PAUSE` den anderen Anwendungen überlassen.

Mit `MULTI` und `CLOCK WAKE` wird der Background-Task `CLOCK` zum Leben erweckt, mit `CLOCK SLEEP` bzw. alternativ mit `SINGLE` wird er angehalten. Wenn nun abermals `CLOCK WAKE` bzw. `MULTI` eingegeben wird, läuft die Uhranzeige weiter.

Auf die Ausgabe über den Lautsprecherport wirkt sich die Anzeige leider negativ aus; der Ton wird arg

verlängert. Das normale `BEEP` (7 `EMIT`) hingegen wird nicht beeinträchtigt. Deshalb habe ich das ursprüngliche 7 `EMIT` für `BEEP` wieder hergestellt, was in der Datei `..\SRC\KERNEL-`

```

1 \ Die Zeit im kooperativen Multitasking oben rechts ausgeben.
2 \ Letzte Ergaenzung: 01/04/96 17:59
3 \ Getestet: F-PC v3.5614, MS-Dos v5.0, 80286
4
5 only forth also definitions
6 code Obranch ( f --- ) \ Verzweigt, wenn das Flag ungleich 0 ist.
7                           \ Das ist die Umkehrung zu ?branch
8                           \ aus ..\src\kernel1.seq.
9   pop cx
10  cx<0 if
11    mov es: ip, 0 [ip]
12    next
13  then
14    add ip, # 2
15  next
16  end-code
17 : unless ( n --- ) \ Ersetzt 0= if.
18   \ Eine Anregung aus Randy Walls Perl.
19   compile Obranch ?>mark ; immediate
20
21 : .clock ( --- ) \ Gibt Stunden, Minuten und Sekunden aus.
22   gettime form-time count 3 - type ;
23
24 user
25 0 value foo \ Die User-Variable wird zum Warten gebraucht.
26 forth
27
28 background: clock ( --- ) \ Gibt staendig die Zeit oben rechts aus.
29   begin
30     pause \ Fuer das kooperative Multitasking wichtig.
31     incr> foo \ Die Zeit soll nur ausgegeben werden,
32               \ wenn foo durch 1000 teilbar ist.
33               \ Das verbraucht nicht zu viel Ressourcen
34               \ mit der Ausgabe der Zeit und F-PC laeuft
35               \ einigermassen fluessig weiter.
36     foo 1000 mod unless
37       at? savecursor cursor-off
38       71 0 at .clock
39       at restcursor
40     then
41     again ;
42
43 only forth also definitions
44
45 ' %beep is beep
46 stattoff \ Schaltet die Statuszeile aus.
47 multi clock wake \ Startet die Uhr.

```

-- **Clock**



Portables Forth: Atari-Portfolio

von Wolfgang Führer

Idastr. 19, D-45663 Recklinghausen, Tel. 02361/891344
wfuehrer@uni-upn.forth-ev.de, CIS 100415,3227

Als er 1986 auf den Markt kam, war er eine kleine Sensation. Ein PC - 20 cm lang, 10.5 cm breit, 2.5 cm hoch und 450 Gramm schwer (80C88 CPU mit 4.9152 MHz) - und doch kompatibel? Zu allem Überfluß auch noch die 5 wichtigsten Applikationen fest im ROM (Textverarbeitung, Tabellenkalkulation *.wks kompatibel, Adressdatenbank, Terminplaner mit Alarm und Taschenrechner).

Klein genug also um ihn überall mit hinzunehmen und "groß" genug um mit allem, zwischendurch anfallenden Arbeiten fertig zu werden.

Stichworte: Portfolio Atari

Als ich 1992 beim Durchblättern einer Atarizeitschrift las, daß es ein FORTH für den Portfolio gibt, war meine Kaufentscheidung gefallen (so wie der Preis: anfangs 850.- dann 350.- im Moment 150.-)

Nachdem ich mir dann noch die Parallelschnittstelle gekauft hatte (seitlich aufzustecken), nebst Übertragungsprogramm zum PC, begann ich mit dem UniForth-Sampler zu programmieren (ist übrigens auch in der KBBS zu finden).

Im Laufe von 1 1/2 Jahren lernte ich nun Forth und schrieb verschiedene Anpassungen für den Portfolio, weil der "Porty" doch nur fast kompatibel ist.

Was an nicht-kompatiblen sofort auffällt, ist natürlich der LCD-Bildschirm, der nur 40*8 Zeichen bzw. 240*64 Punkte umfasst. Außerdem ist

der Ur-Portfolio mit nur 128 KByte RAM ausgestattet, die in RAM-Disk und Arbeitsspeicher aufgeteilt werden müssen (z.B. 64/64). Zum Abspeichern von Programmen und Daten können auch noch Speicherkarten in den Rechner eingeschoben werden. Das interne DOS (DIP-DOS) entspricht nur etwa DOS 2.11 mit einigen Ergänzungen und Erweiterungen. Außerdem ist der Portfolio auf Stromsparen ausgelegt und behandelt einige Interrupts anders, um den Rechner "runterschalten zu können". Davon ist vor allem der Bildschirmrefresh betroffen.

Trotzdem läßt sich gut mit dem Portfolio arbeiten, und man kann ihn wirklich immer dabei haben. Seine Fans schwören jedenfalls auf ihn und haben

auch gerade einen eigenen Club gegründet.

Im Moment portiere ich ZF-Forth auf den Portfolio (vielen Dank an Friedrich Prinz) und bin mit meiner Arbeit eigentlich schon ganz zufrieden (hauptsächlich Kosmetik wegen des kleinen Displays). Ich würde mir allerdings wünschen, daß sich ZF-Forth mit 80 oder sogar 64 KByte Arbeitsspeicher zufrieden geben würde - zur Zeit benötigt es mindestens 104 KByte. Wer kann mir helfen, den Returnstack und die separaten Header 16 oder 32 kByte tiefer zu legen? Da ich ZF auf ca. 24KByte abgespeckt habe, bleibt dann immer noch genügend freier Raum für das Direktory übrig.

Eine Sache, die ich neulich in der KBBS aufgeschnappt habe, interessiert mich ungemein, zumal ich mir auch schon Gedanken in dieser Richtung gemacht habe. Bei bigForth ist es möglich, "Module" abzuspeichern. Also fertig kompilierte Teile des Direktory's. So eine Sache in ZF fände ich super. Wäre das nicht mal etwas, was man in der VD diskutieren und aufarbeiten könnte? (Das mit dem "Aufnehmer" fand doch auch sein Echo.) Etwas platzsparenderes als solche Module kann ich mir für den Porty gar nicht vorstellen. Wenn man außerdem noch die Header wegläßt oder zerstört, kann man fertige Programme weitergeben (verkaufen?) ohne sofort den Quelltext preisgeben zu müssen. Soweit der Anstoß des Monats. □

Fortsetzung von der letzten Seite

2.SEQ in %BEEP schon mal richtig definiert worden war, jedoch unnötiger Weise in ..\SRC\SOUND.SEQ auf NE-WBEEP umgelenkt wird. Die letzte Zeile stellt das ursprüngliche Verhalten von BEEP, nämlich 7 EMIT, wieder her.

Wie der Code zeigt, bin ich doch nicht ganz ohne Assembler ausgekommen. Jedoch kann das Wort UNLESS einfach durch 0= IF dargestellt werden. Als ich vor einiger Zeit mit Perl begann, hatte ich mich für das Unless aus Randy Walls Scriptsprache so begeistert, daß

ich es zum Beispiel in C häufig vermißt habe. Im Forth braucht man aber keine Compiler- oder Steuerworte zu missen, man kann sie sich nachbilden. Ich nahm mir also das IF aus ..\SRC\KERNEL-1.SEQ vor und daraufhin das ?BRANCH. Als Resultat sind OBRANCH und UNLESS entstanden. Ich bin der Meinung, damit kann ich sprechenderen Code schreiben, aber das mag jeder anders sehen.

Zum Schluß möchte ich noch feststellen, daß für Umsteiger oder Leute wie mich, die Forth nur von Zeit zu Zeit

nutzen, die Dokumentation von Richard E. Haskell, "The 4th Course", äußerst interessant ist und über den inneren Aufbau von speziell F-PC einen umfassenden, detaillierten Einblick gibt. Ich fand den Text in CompuServe im Forth-Forum, aber sicher ist dieser auch auf einigen FTP-Servern oder in Mailboxen zu finden.

Wenn in Zukunft neben der Schule und Unix noch Zeit bleibt, werde ich hoffentlich diesmal etwas länger bei resp. mit F-PC verharren. □

cpForth für OS/2

Objektorientierte Implementierung der Programmiersprache FORTH mit Schnittstellen zu C++ und REXX.

von Peter Maier-Schuler

Wiesenstraße 19, 88690 Uhdlingen-Mühlhofen
Telefon/Fax: 07556/5618 eMail: 075565618-0001@t-online.de

cpForth wurde als Werkzeug für die Realisierung von Prüfgeräten sowie Test- und Schulungssimulatoren entwickelt. Es entspricht keinem speziellen FORTH Standard, beinhaltet aber alle wesentlichen FORTH Funktionen. Darüber hinaus stellt cpForth einen 'Grundwortschatz' zur Verfügung, der eine effiziente Softwareentwicklung für Projekte aus den genannten Bereichen ermöglicht.

Stichworte: cpForth C++ OS/2 Prüfgeräte Datenerfassung Simulation

cpForth setzt mindestens einen 386 IBM-PC (oder kompatiblen) und das Betriebssystem OS/2 2.0 oder nachfolgende Versionen voraus. Für eine vollständige Nutzung des Leistungsumfangs von cpForth wird außerdem das IBM CSet++ (IBM's C++ Compiler) - einschließlich des Toolkits - vorausgesetzt.

und/oder - wo erforderlich - als Erweiterung von cpForth in C++ oder REXX.

Zielsetzungen

Zielsetzung von cpForth ist es, die Vorzüge von FORTH für eine effiziente Software Entwicklung unter OS/2 nutzen zu können, ohne die Nachteile

anderer FORTH-Implementierungen in Kauf nehmen zu müssen. Für Anwendungen aus den Bereichen Prüfgeräte sowie Test- und Schulungssimulatoren sind vor allem folgende Eigenschaften von FORTH interessant:

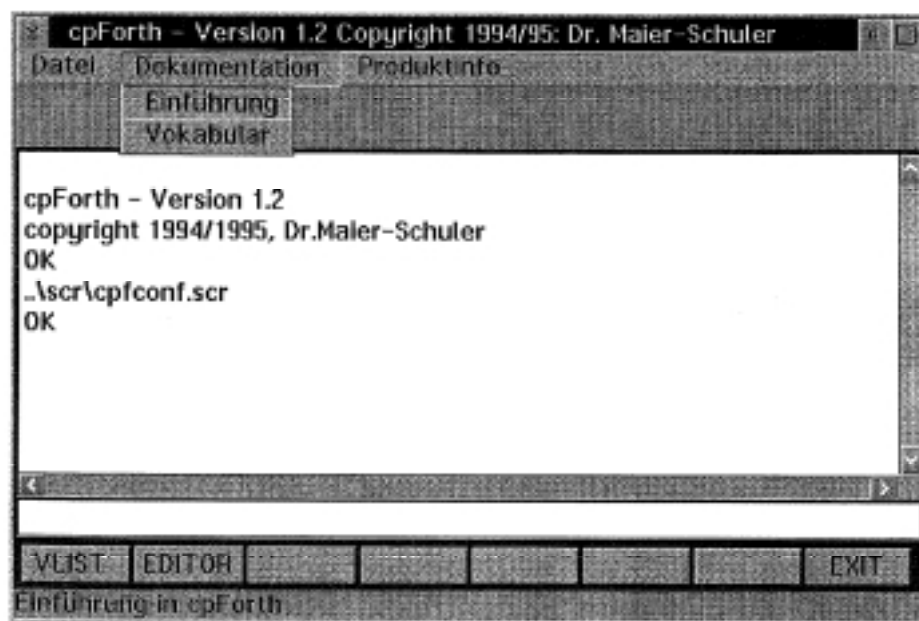
- Codierung nach dem 'bottom-up' Verfahren und sofortige (interaktive) Testbarkeit von kleinen überschaubaren Modulen.
- Die Möglichkeit, für verschiedene Anwendungen spezifische Test- bzw. Schulungssprachen zu entwerfen.
- Durch den interaktiven Interpreter bietet FORTH eine hohe Flexibilität der Anwenderschnittstelle.
- Mit Hilfe interpretierbarer Konfigurationsfiles können leicht verschiedene Softwarekonfigurationen erzeugt werden.

Diesen Vorteilen stehen aber auch einige Nachteile gegenüber, die bei einer Verwendung des Sprachkonzepts von FORTH vermieden werden müssen:

Übersicht

Anwendungen im Bereich Prüf- und Testgeräte besitzen häufig komplexe Anwenderschnittstellen, die leicht an veränderte (erweiterte) Aufgabenstellungen anpaßbar sein müssen. Testabläufe sollen leicht zu erstellen und schnell änderbar sein. Für Prüf- und Testgeräte muß eine einfache Anbindung von PC-Erweiterungskarten und von nachrichtenorientierten Meßgeräten über RS-232 und/oder IEEE.488 Schnittstellen möglich sein. Prüfsimulatoren erfordern darüber hinaus einen zuverlässigen Echtzeittakt, mit dem eine Simulationstask periodisch - in Abständen von ca. 20 - 30 Millisekunden - getriggert werden kann.

Die Elemente dieser Funktionen sind als Bestandteile von cpForth unabhängig von einer speziellen Anwendung realisiert. Die Implementierung konkreter Projekte erfolgt in high-level FORTH



- Leistungsfähige Systeme, welche die FORTH-Philosophie konsequent umsetzen, neigen leicht zu einer, für den Anwender kaum mehr zu überschauenden, Fülle an FORTH-Worten.
- Besonders auf der low-level Ebene und bei der Berechnung von mathematischen Gleichungen wird ein nur schwer lesbarer Code erzeugt.
- Wenig Unterstützung von FORTH durch leistungsfähige Softwarehäuser.
- Der Anwender ist häufig gezwungen, Dinge selbst zu entwickeln, die für andere Programmiersprachen (insbesondere C, C++) in Form von Librarys, Klassenbibliotheken oder anderen Werkzeugen bereitgestellt werden.

Besonderheiten von cpForth

Implementierung: cpForth ist (fast) vollständig in C++ (objektorientiert) implementiert. Der Compiler selbst ist ein C++ Objekt und kann als solches in jede C++ Anwendung integriert werden.

Mit Hilfe verschiedener Klassen kann das Wörterbuch von cpForth direkt in der C++ Anwendung erweitert werden.

Schnittstelle zu C++: Mit Hilfe der FORTH Worte `dll:` und `function:` können C++ Funktionen aus dynamischen `link librarys` (DLL) in FORTH eingebunden werden. Diese Funktionen können vom Anwender selbst entwickelt werden (entsprechende Entwicklungsumgebung vorausgesetzt) und haben direkten Zugriff auf den Datenstack von cpForth.

Kooperatives Multitasking: Beliebige viele cpForth Worte (Tasks) können im Hintergrund ausgeführt werden. Jede Task erhält dabei solange CPU-Zeit, bis sie mit `pause` oder `sleep` freiwillig die Kontrolle abgibt. Bei Verwendung der graphischen Oberfläche von cpForth wird der Output von Background Tasks in anderen Fensterbereichen ausgegeben, als der Output der Foreground Task (default).

Preemptives Multitasking: cpForth stellt einen Timer mit einer minimalen Zykluszeit von einem Systemtick (32 msec) bereit. Damit kann eine sogenannte Synctask zyklisch aufgerufen werden.

cpForth nutzt hier das Multithread Konzept von OS/2 und die Möglichkeit, die Ausführung von Threads über Prioritäten zu steuern. Dadurch ist sichergestellt, daß die Synctask zuverlässig mit jedem Systemtick (oder einem Vielfachen davon) getriggert wird, unabhängig davon, ob die gerade laufende Task bereit ist, die Kontrolle abzugeben.

Bedienoberfläche Der cpForth Compiler und seine Bedienoberfläche sind zwei verschiedene Prozesse. Die beiden Prozesse kommunizieren über eine Pipe miteinander. Sie können daher auch auf verschiedene Rechner verteilt werden. Die Standardoberfläche beinhaltet u.a. eine Menüzeile, mehrere Ausgabefenster sowie eine Leiste mit 'push buttons' (siehe Abbildung). Die buttons können vom Anwender mit eigenen Texten und Funktionen (FORTH Worten) hinterlegt werden.

Durch das Pipe-Konzept ist es leicht möglich, die Standardoberfläche durch eine andere - an entsprechende Projektanforderungen angepaßte - Benutzerschnittstelle zu ersetzen.

Steuerung von Meßgeräten: cpForth unterstützt die Steuerung von nachrichtenbasierten Meßgeräten über serielle Schnittstellen und über den IEEE.488 Bus.

Port I/O: Für einfache Steuerungen (binär Input/Output) ermöglicht cpForth den Zugriff auf die I/O-Ports von PC-Einsteckkarten. Für komplexere Aufgabe (analog Input, Signalverarbeitung etc.) können die vom entsprechenden Hersteller der Karten gelieferten Librarys bzw. Treiber in cpForth eingebunden werden.

Beispiel 1: Multitasking

Das erste Beispiel zeigt die Erstellung von Background Tasks und die Projektierung der Softkey Leiste auf der Bedienoberfläche von cpForth.

Nach dem Laden des nachfolgenden Programms kann über Softkeys ein Zähler als Background Task gestartet, gestoppt und zurückgesetzt werden.

Eine spezielle Hintergrundtask ist die sogenannte 'Synctask'. Die Synctask wird in einem festen Zeittakt von einem oder mehreren Systemticks getriggert.

```

1 ( ***** )
2 (
3 ( cpForth - Testprogramm                Erstellung: 22.1.1995 )
4 ( ===== )
5 ( )
6 ( Testen von Background Tasks          )
7 ( Projektierung der Softkey-Leiste     )
8 (                                     Dr. Maier-Schuler )
9 ( ***** )
10
11 0 variable counts      ( Zählvariable für die Backgroundtask... )
12
13 background: counter   ( Definition einer Backgroundtask )
14   begin
15     1 counts +! counts @    ( Die Ausgabe erfolgt im )
16     ." Zählerstand: " . cr  ( "Backgroundfenster" )
17     500 sleep              ( ca. 500 msec Pause... )
18   again task;
19
20 : counter_start counter start ;
21 : counter_stop  counter stop ;
22 : counter_reset 0 counts ! ;
23
24 4 green button" START" ' counter_start is key_f4 ( Die Softkeys 4, 5 und 6 werden mit )
25 5 red  button" STOP"  ' counter_stop  is key_f5 ( Text und den gewünschten Funktionen)
26 6 black button" RESET" ' counter_reset is key_f6 ( hinterlegt. )
27 ( ***** )

```

Listing 1 : Multitasking



```

1 ( ***** )
2 ( )
3 ( Beispiel für die Erweiterbarkeit von cpForth durch Funktionen aus )
4 ( dynamischen Link Librarys. )
5 ( )
6 ( Voraussetzung: IBM C-Set++ für die Erzeugung von DLL's )
7 ( )
8 ( 19.07.95 Dr. Maier-Schuler )
9 ( ***** )
10
11 dll: ..\usrdll\cmds\mfextent.dll function: inByte ( Schreibweise beachten! )
12 ( Wie im )
13 function: rampe ( C-Quellcode für die DLL. )
14
15 ( in cpForth stehen jetzt die beiden Worte: )
16
17 ( inByte ( n1 -- n1 + 1 ) )
18 ( und rampe ( limit -- amplitude ) )
19
20 ( zur Verfügung (vgl. auch File "mfextent.cpp") )

```



Listing 2 : Erweiterung durch DLL-Funktionen

Sie wird bereits beim Start von cpForth (mit 'noop' initialisiert) gestartet. Mit

```
' <name> is syntask
```

kann der Synctask jedes cpForthWort zugewiesen werden. Über das Laufzeitverhalten (Zykluszeit, Laufzeit etc.) der Synctask geben verschiedene Systemvariablen Auskunft.

Beispiel 2: Erweiterung durch DLL - Funktionen

cpForth kann durch Funktionen aus dynamischen link librarys erweitert werden. Diese Librarys können z.B. mit

Hilfe des C-Set ++ von IBM erstellt werden. Die Funktionen haben Zugriff auf den Datenstack von cpForth, und können somit direkt Daten mit anderen FORTH-Worten austauschen.

Die Code-Sequenz in Listing 2 zeigt, wie eine DLL-Funktion in cpForth eingebunden wird.

Mit `dll: < Pfad\Name>` werden der Pfad und der Namen der Library festgelegt, in welcher sich die nachfolgend verwendeten Funktionen befinden. Mit `function: < Name >` wird ein neues FORTH-Wort "Name" erzeugt, bei dessen Aufruf die DLL-Funktion <Name> ausgeführt wird. Die DLL-Funktion und das FORTH-Wort

besitzen den selben Namen.

Die oben verwendete Funktion 'rampe' kann in C++ z.B. wie in Listing 3 definiert werden.

Der Funktion `rampe ()` wird beim Aufruf in (`MForthStack *`) `dataStack` ein Zeiger auf den jeweils gültigen Datenstack von cpForth übergeben. Mit den Funktionen (genauer: Methoden der Klasse `MForthStack`) `push(x)` und `pop()` können Werte auf den Stack geschrieben, bzw. von dort geholt werden.

Mit dieser einfach zu handhabenden Schnittstelle kann cpForth durch beliebige C++ - Funktionen (z.B. zur Berechnung mathematischer Gleichungen, Simulationsmodellen etc.) erweitert werden.

Ausblick

cpForth unterstützt die Entwicklung von Anwendungen mit Schwerpunkten auf der Datenerfassung und der Automatisierung von Prüf- und Testabläufen. Wie die ausgeführten Beispiele zeigen, orientiert sich an dieser Zielsetzung der von cpForth bereitgestellte Sprachumfang. Vorgefertigte Lösungen in Form von leistungsfähigen FORTH Worten und offene Schnittstellen gewährleisten eine effiziente Anwendungsentwicklung.

cpForth wird laufend weiterentwickelt und neuen Anforderungen angepaßt.

Geplant sind folgende Erweiterungen:

- Zahlendarstellung/Arithmetik in floating point: 'Überladene' Operatoren für integer- und floatingpoint Arithmetik.
- Typprüfung(int, float, Adresse) bei Stackoperationen zur weiteren Verbesserung des Fehlerhandlings.
- DDE-Schnittstelle zu anderen OS/2- und WINDOWS-Applikationen.
- Kommunikations-Schnittstellen über pipe und shared-memory.

Damit erhält cpForth die Fähigkeit, unterschiedlichste Softwarepakete (Tabellenkalkulation, Telekommunikation, Prozeßvisualisierung ...) zu integrieren und deren Funktionalität zu nutzen. □

```

1 #include "..\..\cpf83\src\mfstack.hpp" /* Include der Klasse MForthStack */
2
3
4 int inByte(MForthStack *); /* Prototyp deklarieren */
5
6 long int rampenwert = 0; /* Globale Variable: Wert bleibt */
7 /* zwischen zwei Aufrufen von */
8 int rampe(MForthStack *dataStack) /* rampe() erhalten. */
9 {
10     long int maxwert;
11
12     maxwert = dataStack->pop(); /* maxwert vom cpForth Stack holen*/
13     rampenwert += 1; /* neuen Rampenwert berechne... */
14     if(rampenwert >= maxwert) rampenwert = -maxwert; /* .... */
15     dataStack->push(rampenwert); /* und auf den cpForth Datenstack */
16 /* schreiben. */
17     return(0);
18 }
19

```



Listing 3 : Definition von Funktionen in C++

Forth International

von Fred Behringer



Schweden: 32-Bit- TILE-Forth

Es gibt eine CD mit dem Titel "X11R5 GNU" von Walnut Creek (December 1992). DM 39,-. Auf dieser habe ich das (T)hreaded-(I)nterpretive-(L)anguage-(E)nvironment-Forth entdeckt.

Autor: Mikael R.K. Patel; Computer Aided Design Laboratory (CADLAB); Department of Computer and Information Science; Linköping University; S-581 83 LINKÖPING SWEDEN; Email: mip@iPda.liu.se

Die Version ist 2.1 vom August 1990. Es handelt sich um eine 32-Bit-Implementation von F83, die als portierbares

Paket für Workstations und "mittelgroße" Computer gedacht ist. Die in C geschriebenen Quelltexte füllen insgesamt (nur) etwa 350 KB. Dazu kommen nochmal 200 KB Dokumentationen (UNIX-man-pages usw.). Der Autor arbeitet mit SUN-Maschinen. Das Ganze ist auf den GNU-C-Compiler zugeschnitten.

Auf besagter CD scheint die gesamte GNU-Software vom Stand 1992 zu liegen. GNU = (G)nu is (N)ot (U)nix. TILE-Forth liegt im Verzeichnis gnu/tile-forth-2.1.

Den GNU-CC-Compiler gibt es auf dem Public-Domain-Markt (zu den Bedingungen der Free Software Foundati-

on) in verschiedenen Ausführungen: Für DOS-486-Maschinen, für OS/2 und für LINUX. Die DOS-Version arbeitet mit einem DOS-Extender und erzeugt ausführbare EXE-Dateien. TILE-Forth müßte sich also auch für die DOS-Ebene compilieren lassen.

Die CD läßt sich unter DOS (mit den üblichen Schwierigkeiten: kein Linefeed) lesen. Besser geht es natürlich, wenn man sie sich sofort unter LINUX ansieht.

Der Autor erlaubt die kostenlose Weitergabe. Bei Dauerbenutzung erbittet er eine "Donation" von 50 Dollar.

Frankreich: Turbo- Forth-Auskünfte jetzt auch über INTERNET

Marc Petremann, der Initiator, Hauptentwickler und Vertreiber von Turbo-Forth, ist jetzt auch per e-mail zu erreichen:

100647.3306@compuserve.com

Fred Behringer, München

Gehaltvolles

zusammengestellt und übertragen
von Fred Behringer

Forth Dimensions der Forth Interest Group, USA

September/Oktober 1995

6 Sets, Stacks, and Queues

Marty McGowan

Was läßt sich über Stacks noch sagen? Dieser Artikel beschäftigt sich weniger mit Gleitkomma-, Compiler- oder Ausnahmebehandlungs-Stacks als vielmehr mit der Verwendung von Stacks in Software-Anwendungen, d.h., Stacks in etwas allgemeinerem Zusammenhang mit "sets" und "queues". Diese drei Methoden unterscheiden sich nur in der Art des Zugriffs: LIFO, FIFO und "AIRO". Macht man sich mit Forth-Verwirklichungen dieser drei Methoden vertraut, so fällt es dann in den Anwendungen leicht, die jeweils passende Methode einzusetzen.

14 Bounds Checking for Stacks

Eine Diskussion, die in der Internet-Abteilung comp.lang.forth von Russell Y.

Webb entfacht wurde. Sie dreht sich einerseits um eine interessante Verfahrensfrage, wirft aber auch gleichzeitig Licht auf das Lösen von Problemen in Forth ganz allgemein. Es fing alles mit einer unschuldigen kleinen Online-Bitte um Rat an: "Was ist die wirksamste Methode zur Überprüfung von Stack-Überlauf und -Unterlauf? ... Ich bin an einer hinreichend sicheren, stack-orientierten virtuellen Maschine interessiert, aber es scheint eine Menge Verwaltungsaufwand nötig zu sein, um alles durchzuprüfen. Ich bin für jeden Hinweis dankbar."

20 Nanocomputer Optimizing Target Compiler: the Processor-Independent Core

Tim Hendtlass

Mit schöner Regelmäßigkeit erscheinen neue Nanocomputer - kleine Einchip-Prozessoren mit integriertem RAM, ROM und E/A - und eine einfache Alternative zur Maschinensprache kann die Entwicklung von Anwendungen für solche Prozessoren beschleunigen. Der hier beschriebene prozessor-unabhängige Kern braucht nur auf eine prozessor-spezifische Bibliothek abgestimmt zu werden, um einen Compiler zu liefern, der Forth als Eingabe akzeptiert und absoluten Maschinencode erzeugt. (In der nächsten Ausgabe bringen wir Bibliotheken für den

PIC16C71 und den PIC16C84.) Der Compiler kann für Chips von unterschiedlicher Wortlänge und unterschiedlichem Aufbau verwendet werden. Vom Zielprozessor wird lediglich erwartet, daß er eine Folge von Befehlen aus irgendeiner Art von ROM aufnehmen kann und etwas RAM zum Ablegen von Variablen und zur Einrichtung von Stacks hat. □

Ich möchte in dieser Rubrik in unregelmäßiger Folge über Forth-Aktivitäten in anderen Ländern berichten. Ich bin fleißig am Sammeln, würde mich aber natürlich auch über Zusendung von Material von anderer Seite freuen.

Fred Behringer
Planegger Str. 24
81241 München.

**Verständlich sind für mich die
Sprachen Englisch, Holländisch, Ita-
lienisch und Französisch.**



So, hier liegt sie nun - die CD Forthige Weihnacht 95 - genug Schweiß hat sie alle Mitwirkenden wohl gekostet, und wir alle haben eine Menge gelernt. Was da alles zu beachten ist, wenn man über die Republik verteilt an der Zusammenstellung solcher Datenmengen arbeitet. Ich sage nur 'Kopfstierschraube'. Die CD, als Abzug der beiden Forth FTP-Server Taygeta und Asterix zusammen mit einer losen Zusammenstellung von mehr oder minder sortierten Quellcodesammlungen verschiedener Autoren ist eine richtige Fundgrube. Sie enthält auch alles, was mensch zum ANS-Forth Programmieren braucht.

Alle von mir das letzte Mal besprochenen frei verfügbaren ANS-Forth-Systeme sind auf der CD zu finden und können problemlos von dort aus installiert werden. Auch das von mir noch im letzten Jahr in einer neu überarbeiteten und erweiterten Version fertiggestellte

ANS-Forth Kompatibilitäts-Paket für F-PC findet sich dort und kann zusammen mit dem auf der CD installierten F-PC mit wenigen Handgriffen als Basis zur Entwicklung von Standardprogrammen eingesetzt werden.

Das letzte mal habe ich über die Diskussion in *de.comp.lang.forth* berich-

tet, wie die Standardbegriffe geeignet in die deutsche Sprache übersetzt werden können oder sollten. Da keine weiteren Diskussionsbeiträge mehr eingegangen sind, ist es wohl Zeit, den status quo hier ersteinmal wiederzugeben

Jetzt
geht's
ANS
FORTH!

von Ulrich Hoffmann



Deutschsprachige Übersetzung der ANS-Forth Begriffe (unvollständig)

AE: Anton Ertl	character string	current definition
WOI: Ehmont Woitzel	AE: Zeichenfolge (oder -string)	AE: aktuelle Definition
UHO: Ulrich Hoffmann	WOI: Zeichenkette	data field
RH: Rainer Hansen	code space	WOI: Datenfeld
CLV: Claus Vogt	AE: Code-Fläche (Ausfuehrungsfläche?)	UHO: Datenfeld
address unit	CLV: Codebereich	data space
AE: Adressierungseinheit	WOI: Programmraum	AE: Datenfläche
WOI: Adreßeinheit (kürzer)	UHO: Programmbereich	CLV: Datenbereich
UHO: Adreßeinheit	compile	WOI: Datenraum
aligned address	AE: übersetzen	UHO: Datenbereich
AE: (Zellen-)ausgerichtete Adresse	CLV: kompilieren	data space pointer
ambiguous condition	WOI: compilieren	AE: DP(-)
AE: mehrdeutiger Zustand (undefinierter Zustand?)	UHO: übersetzen	WOI: Datenraumzeiger
CLV: Mehrdeutigkeit, Fehlerbedingung (volksForth83)	compilation semantics	UHO: Datenbereichs-Zeiger
WOI: mehrdeutige Situation (ist nicht ganz so statisch)	AE: Übersetzungsbedeutung (oder -semantik?)	data stack
UHO: unspezifizierte Situation	CLV: Kompilationsverhalten	AE: Datenstapel (oder -stack oder -keller)
cell	WOI: Compilations-Semantik	data type
H: Zelle	counted string	AE: Datentyp
AE: Zelle	AE: nicht übersetzen, das sollte man eh nicht verwenden	defining word
cell pair	WOI: abgezählte Kette (wenn schon, denn schon)	AE: Definitionswort
AE: Zellenpaar	UHO: counted string	UHO: definierendes Wort
character	cross compiler	definition
AE: Zeichen	AE: Quer-Übersetzer:-(AE: Definition
character-aligned	CLV: nicht übersetzen	dictionary
AE: Zeichen-ausgerichtet	WOI: Crosscompiler (Kreuzworträtsel-Maschine?)	AE: Wörterbuch (aber: Vocabulary - Wortliste)
	UHO: Crosscompiler	



display

AE: Anzeige

environmental

AE: Umgebungs-

dependencies

AE: Abhängigkeiten

execution semantics

AE: Ausführungsbedeutung/-semantik

WOI: Ausführungssemantik

UHO: Ausführungssemantik

execution token

AE: Execution Token (Wort-ID?)

CLV: Ausführungstoken

WOI: Ausführungs-Token

UHO: Token

find

AE: suchen oder finden

UHO: evtl. aufsuchen oder auffinden

immediate word

AE: sofort (während der Übersetzung) ausgeführtes Wort

CLV: unmittelbares Wort (?) - besser unübersetzt lassen

WOI: bevorzugtes Wort

UHO: bevorzugtes Wort

implementation defined

AE: von der Implementation definiert (bzw. zu definieren)

WOI: implementationsdefiniert

implementation dependent

AE: implementationsabhängig

input buffer

AE: Eingabebereich

CLV: Eingabepuffer

WOI: Eingabepuffer

UHO: Eingabepuffer

input source

AE: Eingabequelle

input source specification

CLV: Eingabemedium / spezifizierte Eingabequelle

WOI: Eingabequellen-Spezifikation

UHO: Eingabequellen-Angabe

interpretation semantics

AE: Interpretationsbedeutung (oder -semantik)

WOI: Interpretationssemantik

UHO: Interpretationssemantik

keyboard event

AE: Tastaturereignis

line

AE: Zeile

name space

AE: Namensfläche

CLV: Namensbereich

WOI: Namensraum

UHO: Namensbereich

number

AE: Zahl

parse

AE: (auf)trennen

CLV: analysieren, durchsuchen (?)

WOI: parsen

UHO: zerlegen, parsen

parse area

AE: Trennbereich

WOI: Parse-Bereich

pictured-numeric output

AE: (speziell) formatierte Zahlenausgabe

WOI: bildhafte numerische Ausgabe (absolut häßlich)

UHO: formatierte Zahlenausgabe

program

AE: Programm

receive

AE: erhalten

WOI: empfangen

UHO: empfangen

return stack

AE: Rücksprungstapel (...)

CLV: Rückkehrkeller :)

WOI: Rückkehrstapel

UHO: Rückkehradressen-Keller

standard word

AE: Standard-Wort

user input device

AE: Benutzereingabegerät

WOI: Anwender-Eingabegerät

user output device

AE: Benutzerausgabegerät

WOI: Anwender-Ausgabegerät

variable

AE: Variable

word

AE: Wort

word list

AE: Wortliste (oder Wörterliste?)

word set:

AE: Wortmenge (oder Wörtermenge?)

CLV: Wortgruppe

WOI: Wortschatz

UHO: Wortgruppe



Der von Rick VanNorman in comp.lang.forth ausgeschriebene ANS-Forth Programmierwettbewerb

“Wer Programmiert SOKOBAN?”

ist mittlerweile abgeschlossen. Rick hat die eingegangenen Lösungsvorschläge unter Zustimmung ihrer Autoren in comp.lang.forth veröffentlicht. Interessant ist zu beobachten, daß die Verwendung des Standards offenbar nicht einen speziellen Forth-Stil erzwingt. Die Stilvariation, die man in den unterschiedlichen Lösungen findet belegen dies.

Auf der technischen Schiene wird in (de.)comp.lang.forth gerade über so wichtige Dinge diskutiert, wie die Frage, ob denn “-.” dem Standard nach eine Zahl sei oder nicht. Das ist eine Frage, die für System-Implementierer

natürlich von Bedeutung ist, da von ihr die Implementierung der Konvertierung aus der externen in die interne Zahlendarstellung abhängt. Für Forth-Programmierer, die portable Programme schreiben wollen, ist dies aber kein ernstzunehmendes Problem. Soll “-.” in ihrem Programm eine doppeltgenaue 0 sein, so definieren sie am Anfang ihres Programms:

```
: -. ( -- 0. ) 0. ;
```

Wollen sie jedoch “-.” als Zahl verbieten, schreiben sie einfach:

```
: -. ( i*x -- )
  ." -. is not a valid number!"
  ABORT ; IMMEDIATE
```

Die Frage, ob denn nach Laden eines solchen Programms auf ein Standard-System denn noch immer ein Standard-System existiert - eine der vom Standard geforderten Mindestangaben zu einem Standard-Programm - bleibt zwar schwierig generell zu beantworten, den speziellen Sachverhalt sollte man aber wohl schon darlegen können.

Viele Grüße und bis zum nächsten Mal.

May the Forth be with you!



Forth auf eine professionelle Basis heben

von Peter Knaggs, Paisley, Schottland

pjk@paisley.ac.uk

Der Abdruck dieses Artikels erfolgt mit freundlicher Genehmigung der amerikanischen Forth Interest Group. Der ursprüngliche Artikel erschien in der "Forth Dimensions" vom November/ Dezember 1995. Die Übersetzung für das Magazin "Vierte Dimension" besorgte Fred Behringer, München.

Stichworte: Diskussion Professionell

Forth hat bisher unter der Auffassung gelitten (und leidet immer noch darunter), es sei eine Sprache für Hacker. Das geht bis in die Tage zurück, als FIG-Forth herausgegeben wurde. Softwarehäuser und deren Manager warfen einen kurzen Blick auf dieses Durcheinander von Ideen und schlecht

zu einer anerkannten Programmiersprache heranwachsen will. Und sie wird sich ändern. Und sie hat sich auch schon geändert. ANS-Forth und die IEEE-Open-Firmware-Bemühungen sind große Schritte in dieser Richtung. Sie verhalfen Forth allmählich zu einem gewissen Ansehen.

Wenn wir wollen, daß Forth ernstgenommen wird, dann sollte es einen Fachverband geben, der die Qualität von Forth-Programmierern

dokumentierten Programmen, die keinem Standard genügten, und wurden entweder Enthusiasten oder ließen es wie eine heiße Kartoffel fallen. Wenn heutzutage von Forth gesprochen wird, dann ist das das einzige, an was man sich erinnert. Wir wissen alle, daß sich Forth seither meilenweit nach vorn bewegt hat, doch in deren Vorstellung ist es gerade noch so wie damals.

Software-Manager neigen dazu, nur auf den (meist schlecht dokumentierten) Code zu achten. Sie begreifen nicht, daß es sich nicht einfach nur um noch eine Sprache handelt, sondern daß eine ganz bestimmte Programmier-Philosophie dahintersteckt. Viele Ideen, die heutzutage groß geschrieben werden (strukturierte Programmierung, Wiederverwendbarkeit, Bibliotheken usw.) und die man unter "Software-Engineering" zusammenfaßt, werden in Forth schon seit Jahren verwendet.

Die genannte Einstellung muß sich ändern, wenn Forth in irgendeiner Weise

Von verschiedenen Organisationen wurde eine ganze Reihe von maschinenunabhängigen Entwicklungssprachen geschaffen: Die Ten15-Development-Language (TDL), Architecture-Neutral-Distribution-Format (ANDF), Open Firmware und JAVA, alles Sprachen, die eine forth-artige abstrakte Maschine verwenden, um portierbare Programmierung auf verschiedenen Ebenen zu ermöglichen.

Der ANSI-Standard hat sich schon sehr bemüht, das Ansehen von Forth in den Augen vieler Manager zu heben. Der Portabilitätsaspekt war dabei sehr wichtig. Der ANSI-Standard spricht das Problem nicht einfach nur an, er erleichtert direkt die Entwicklung portierbarer Programme.

Das Eindringen von Forth in die professionellen Programmierstuben wird nicht nur durch den Mangel an guten Forth-Programmierern erschwert, sondern auch durch das Zögern der Softwarehäuser, die Verwendung von

Forth als Programmiersprache zu würdigen.

Der Open-Firmware-Standard hat eine Nachfrage nach versierten Forth-Programmierern geschaffen, der wir einfach nicht gerecht werden können.

Eine immer größere Zahl von Firmen verwendet Forth, aber scheut sich, das zuzugeben. Zwei Gründe sind dafür bestimmend: Einerseits fürchten sie, der Markt könne sensibel reagieren, andererseits ziehen die Manager in Betracht, daß man sie vielleicht für eine solche, offensichtlich intelligente Fertigungsentcheidung verurteilen würde.

Das wissenschaftliche Interesse an Forth, sei es als Sprache an sich, sei es als Entwicklungsplattform, leidet darunter, daß es kein normales Gutachterverfahren gibt. Momentan bieten sich wenig Möglichkeiten für wissenschaftliche Veröffentlichungen. Eigentlich gibt es für solche Arbeiten nur zwei Wege:

Die euroForth-Konferenz hat in ihren Tagungsberichten kürzlich einen Abschnitt "begutachtete Arbeiten" eingerichtet. Keine der anderen Forth-Veranstaltungen bietet etwas Derartiges. Die einzige andere Möglichkeit ist das Journal of Forth Application and Research, das sein Erscheinen erst kürzlich, nach einer Pause von fünf Jahren, wiederaufgenommen hat. Die anderen forth-bezogenen Publikationsorgane arbeiten nicht nach dem Gutachterprinzip.

Die Situation hat sich seit 1993, als noch nichts dergleichen zur Verfügung stand, enorm verbessert. Trotzdem bleibt noch ein weiter Weg, bevor Forth in der Fachwelt akzeptiert wird.

Ich bin der festen Überzeugung, daß FIG-Forth, oder genauer gesagt, das Belassen von FIG-Forth in seinem damaligen Zustand, sehr viel zu dem Vorurteil beigetragen hat, das uns heutzutage entgegengebracht wird. Sowohl die Forth Interest Group (FIG) als auch Forth selbst werden ständig mit der Haltung von "Hackern" in Verbindung gebracht.

1991, während der Forth-Konferenz in Rochester, habe ich an einer Reihe von Diskussionen mit den verschiedensten Leuten teilgenommen, Leute von FIG (mit Vorsitzendem und Direktoren) und solche des Institute of Applied Forth Research (IAFR). Ich habe mir diese

Diskussionen durch den Kopf gehen lassen und bin zu folgenden Überlegungen gelangt.

Wenn wir wollen, daß Forth ernstgenommen wird, dann müssen wir uns einen Fachverband zulegen, eine Einrichtung, die die Qualität von Forth-Programmierern einstuft und überwacht. Das halte ich für sehr wesentlich. Denn nur über eine solche offizielle Einrichtung können wir es schaffen, das Klischee vom "Hacker" loszuwerden. Lassen Sie mich diese Einrichtung für den Rest der vorliegenden Arbeit den "Fachverband der Forth-Programmierer" nennen.

Ich stelle mir diesen Fachverband als eine Organisation vor, die von Mitgliedern getragen wird, ein Verband zur Pflege, Überwachung und Förderung guter Programmierpraxis in Forth-Umgebungen. Über die genaue Art eines solchen Verbandes mache ich mir noch keine Gedanken. Ich möchte aber kurz ein paar Punkte aufzählen, denen er bestimmt genügen muß, wenn er die oben angedeuteten Ziel erreichen soll:

- Der Fachverband würde auf Subskriptionsbasis aufgebaut sein, mit verschiedenen Arten von Zugehörigkeit, je nachdem, welche Fähigkeiten der einzelne in bezug auf professionelle Software-Entwicklung in Forth mitbringt. Man könnte Statusbezeichnungen festlegen, wie zum Beispiel "Lernender", "Zahlendes Mitglied", "Förderndes Mitglied", "Mitglied", "Beratendes Mitglied".

Ein wohldefiniertes System vom Kriterien würde die Kompetenzstärke auf der jeweiligen Mitgliedschaftsstufe regeln. Die Mitglieder wüßten dann genau, wo sie noch an sich arbeiten müßten, um in der Skala der Mitgliedschaften aufzusteigen.

Die Firmen könnten sich dann ein besseres Bild von dem Bewerber, den sie anstellen wollen, und dessen Fähigkeiten machen.

- Es gibt zur Zeit eine ganze Reihe von Institutionen und Organisationen, die Forth-Kurse veranstalten. Der Fachverband könnte nach und nach diesen Bemühungen einen offiziellen Anstrich geben. Leute, die solch einen Kurs besuchen, wären dann automatisch Kandidaten für den Status eines "Lernenden

Mitglieds". Und nach Absolvierung eines solchen Kurses (solcher Kurse) könnte man ihnen dann genügend Wissen und Erfahrung bescheinigen, um sie auf die Stufe eines "Mitglieds" zu heben.

Firmen oder Einrichtungen, die solche Kurse veranstalten, könnten diese dann in ihren Anzeigen als "vom Fachverband anerkannt" anpreisen. Das würde den Wert der Kurse steigern, aber auch zur Profilierung des Fachverbandes beitragen.

- Um die Entwicklung auf dem Gebiet der Forth-Programmierung anzukurbeln, könnte der Fachverband regelmäßige internationale Treffen mit nach dem Gutachterprinzip einzureichenden Redebeiträgen organisieren. Zur Förderung der sozialen Kontakte ließen sich informelle Zusammenkünfte auf nationaler oder regionaler Ebene einrichten.

Der Fachverband sollte auch ein Forum für tiefere Arbeiten schaffen. Er sollte eine auf Gutachterbasis gegründete Zeitschrift wissenschaftlichen Anspruchs über Entwicklung, Praxis und Erfahrungen im Umgang mit Forth herausgeben. Über die Häufigkeit des Erscheinens ließe sich reden, vielleicht sogar nur einmal pro Jahr. Andererseits, wie bei den informellen örtlichen Zusammenkünften auch, wäre zusätzlich noch eine leichter verdauliche Zeitschrift, Forth-Spektrum oder dergleichen, angebracht, die regelmäßiger, sagen wir monatlich erscheinen könnte und zur Mitteilung von Neuigkeiten, für Produkt- und Stellenanzeigen und andere mehr gesellschaftliche Dinge verwendet werden könnte.

Wie man sieht, ist der Aufgabenbereich des vorgeschlagenen Fachverbandes doch recht groß und umfaßt die derzeitigen Aktivitäten sowohl von FIG als auch von IAFR. Wie der genaue Aufbau des Fachverbandes aussehen soll, ist eine andere Frage.

Man könnte eine ganz neue Organisation ins Leben rufen. Diese müßte aber in direkte Konkurrenz zu FIG und IAFR treten. Sicher wäre das eine "marktgerechte" Lösung. Es würde aber die Forth-Gemeinde nur noch weiter aufspalten. Es würde eine Zeit lang zu

doppelt produzierten Anstrengungen führen, solange, bis sich der eine oder andere der drei Konkurrenten zurückzieht. Keine guten Aussichten.

Eine andere Möglichkeit wäre, unter der Schirmherrschaft einer schon bestehenden "professionellen" Organisation eine spezielle Fachgruppe einzurichten. In Frage kämen BCS, ACM, IEE oder sogar auch IEEE. Dabei fällt mir die inzwischen eingegangene ACM-SIG-Forth ein, wenn diese auch nicht so aktiv war. Unter Umständen ließe sich etwas Geld aus dem Open-Firmware-Lager für ein solches Vorhaben lockermachen.

Noch eine andere Möglichkeit wäre natürlich auch, FIG oder IAFR zu erweitern und die Aktivitäten des "neuen" Fachverbandes übernehmen zu lassen.

Schließlich bestünde auch noch die Möglichkeit, FIG und IAFR zu einem einzigen Verband der vorgeschlagenen Art zu verschmelzen. Das hätte dann jedoch den Nachteil, daß die Hackerhaltung, die man FIG nachsagt, auf die neuzugründende Organisation überginge. Es wäre vernünftiger, FIG in IAFR aufgehen zu lassen.

Ich stelle die hier dargelegten Ideen zur Diskussion. Einige davon finden sich in weiter ausgearbeiteter Form in meiner, für euroForth '93 verfaßten, Arbeit "A Look at Forth's Academic Standing". Diese Arbeit ist auch über folgende Quelle zu beziehen:

<URK:ftp://ftp.paisley.ac.uk/pub/cis/forth/ef93/academic.ps>

oder über das World Wide Web unter:

<URL:http://www.paisley.ac.uk/cis/forth/ef93/academic.html>



Peter Knaggs ist Dozent für Software-Engineering an der Universität von Paisley, Schottland. Er promovierte 1994 an der Universität von Teesside mit einer Arbeit über "Praktische und theoretische Gesichtspunkte zur Software-Entwicklung unter Forth". Er ist aktiv mit forthbezogenen Forschungen beschäftigt und leitet das Programm-Komitee der European Forth Conference (euroForth). Außerdem verwaltet er die euroForth WWW site:

<URL: http://www.paisley.ac.uk/~cis/forth>

Forth braucht keinen qualitätsüberwachenden Fachverband

von Fred Behringer

Planegger Str. 24; D-81241 München

Versuch einer Antwort eines "Unprofessionellen" auf den Artikel "Making Forth Professional" von Peter Knaggs, Schottland, in der "Forth Dimensions" vom November/Dezember 1995. Eine Übersetzung des Artikels von Peter Knaggs ins Deutsche findet sich im vorliegenden Heft der "Vierten Dimension". Anknüpfungspunkt ist im Zweifelsfall stets das Original.

Stichworte: Professionell Diskussion Hobby

Bekenntnis zu Forth

Es ist nicht Forth selbst, es ist das Geheimnisvolle an Forth, das mich fasziniert: Auf den ersten Blick scheint alles möglich. Erst wenn man sich näher mit den Dingen beschäftigt, merkt man, daß es doch nicht ganz so einfach ist. Beispiel: Ein interaktiver Druckertreiber für den NEC-P60. In Forth überhaupt kein Problem. Sofort hingesetzt und gemacht. Nach einem halben Jahr war ich dann schlauer. Heute würde ich dasselbe Projekt gleich von vornherein systematischer anpacken. Aber gelernt habe ich in kürzester Zeit unendlich viel. Ganz ohne Lehrgänge und Kurse (in irgendeinem "Fachverband"). Aber natürlich in unzähligen begeisterungsvollen Nächten am Bildschirm. Forth professionalisieren? Nein. Laßt Forth ruhig eine geheimnisvolle, von vielen als exotisch verschrieene Sprache bleiben!

"Forth ist überhaupt keine Sprache"? OK. Dann erklärt aber diese Sprache doch bitteschön offiziell zur Nichtsprache und laßt sie mir! Von allen Nichtsprachen, mit denen ich mich beschäftigt habe, ist mir Forth die liebste.

In der Physik und im Wirtschaftsleben, und nicht nur dort, kennt man neben geschlossenen Systemen auch offene. Forth ist ein offenes System, ein grenzenlos erweiterbares. Was ist eigentlich gegen offene Systeme einzu-

wenden? Der Systematiker möchte die Welt gern in ein geschlossenes System zwingen. Es muß doch aber auch Menschen geben dürfen, die offene Systeme lieben.

Soll man Forth sprachliche Bandagen anlegen?

Ich habe vor Jahren mal spaßeshalber ein Programm zur Überprüfung der Verteilung von Primzahlwillingen geschrieben. In Forth überhaupt kein Problem. Und es fallen einem laufend Tricks ein, wie man die Wirksamkeit eines solchen Programms steigern kann - wenn man seine Maschine kennt. In anderen Sprachen nützt mir die Maschinenkenntnis weniger. Ich weiß ja gar nicht, wie der jeweilige Compiler das oder jenes umsetzt und wo er das oder jenes hinlegt. Forth beläßt mir die Maschinennähe und befreit mich gleichzeitig von lästiger Verwaltungsarbeit.

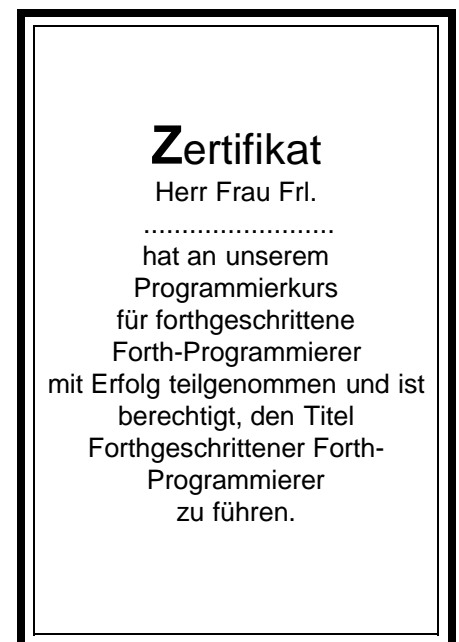
Im Verlaufe einer Vorlesung über Zuverlässigkeitstheorie wollte ich eines Tages mal ganz schnell an einem Beispiel zeigen, daß eine bestimmte Art der Abschätzung nichts mit einer bestimmten anderen Art zu tun hat. Hierzu war das Aufspüren von kürzesten Wegen in einer Weg-Schnitt-Darstellung der Zuverlässigkeitsfunktion angesagt. Pseudoboolesche Funktionen in einer IF-Verbindung. Kein wirkliches Problem für Forth. An einem Abend erledigt. Machen Sie das mal in Turbo-

PASCAL oder OCCAM. Wie umständlich! Und ineffizient! Und dauernd blickt einem jemand über die Schulter und bemerkt "ja das geht in Turbo-PASCAL soundsoviel nicht, versuchen Sie es mal mit Version soundsoviel". An den Borland-Compiler komme ich nicht ran - wenn ich nicht endgültig ins Hackerlager überwechseln möchte. In Forth weiß ich, was ich tue.

Laßt Forth so offen, wie es ist! Auch wenn manche höhnen, Forth sei keine Sprache, Forth sei eine Krankheit. Denken wir denn, wenn wir denken, immer in sprachlichen Kategorien? Das wäre ja schlimm! Muß es denn beim Programmieren immer gleich eine abgeschlossene, metasprachlich faßbare Fix- und-fertig-Sprache sein?

Soll man Forth den "Programmierern" überlassen?

Brötchen backt der Bäcker. Was ist ein Programmierer? Einer, der im Auftrag eines Arbeitgebers für Geld einen gewissen Programmablauf herstellt, der eine Allzweckmaschine befähigt, gewis-



se Eingangssignale in gewisse Ausgangssignale umzuwandeln. Die Mathematiker sprechen von einem Algorithmus - vorausgesetzt, die Signale lassen sich in Zahlen ausdrücken. Ein Programmierer setzt einen ihm vorgegebenen



Logophilie im Anfangsstadium

Algorithmus in irgendeine maschinenverständliche Sprache um. Was hat das aber mit Forth zu tun? Wieso brauchen wir eine Fachvertretung für Forth-Programmierer? Gibt es einen Fachverband für Turbo-PASCAL-Programmierer? Sowas macht man einfach. Dazu braucht man keinen Verband. Ingenieure, Techniker, Naturwissenschaftler, Leute aus dem Wirtschaftsleben, Entscheidungsträger, Linguisten, alle stellen laufend "Programme" her. Sind das jetzt schon Programmierer, die sich evaluieren lassen müssen?

Nein, sie sind weniger als das. Sie sind keine professionellen Programmierer. Sie sind aber andererseits auch sehr viel mehr. Viele von uns sprechen mehrere natürliche Sprachen, ohne deswegen gleich Philologen zu sein. Trotzdem kann es aber, und darf es doch, vorkommen, daß man sich zum Liebhaber derjenigen Sprachen entwickelt, die man spricht, zum "Logophilen".

Soll man sich die Freude am Umgang mit Forth dadurch nehmen lassen, daß man sich in die Kaste der Programmierer zwängen läßt? Soll man sich von irgendeiner Gilde einen Gütestempel aufdrücken lassen? Wozu? Der Verband der Programmierer - auch der Forth-Programmierer - interessiert mich wenig, Forth tut das sehr wohl.

Nichts gegen "Professionelle". Sie haben ihre Daseinsberechtigung. Gerade wir in Deutschland sind stolz darauf, daß jeder Berufsausübende seinen Beruf als Berufung auffaßt und sein Bestes und nur sein Bestes gibt. Andererseits

sollte man aber auch eine Tür für den begeisterten Anfänger offenlassen, für den interessierten Laien, der auf diese Weise aus seinem schulzeugnisgeprägten Fachidiotentum ausbrechen möchte. Mir ist jedenfalls ein forth-übernächtiger Enthusiast, der mit unzulänglichen Mitteln unmögliche Fragen zu beantworten sucht, lieber als ein ausgelaugter "Professioneller". Halte Forth offen für die Begeisterungsfähigen! Nirgendwo sonst können sie sich so zu Hause fühlen wie in Forth.

Kann man für Forth die Akzeptanz erzwingen?

Was ist der Unterschied zwischen einer natürlichen Sprache und einer künstlichen? Die natürliche hat so viele Redundanzen eingebaut, daß ein Wirtschaftskraker Volksreden halten kann, ohne jemals Reden gelernt zu haben, und dennoch verstanden wird. Bei künstlichen Sprachen geht das, oder ging das bisher, nicht. Das ist alles. Welche Sprache wird im Europa der Zukunft vorherrschend sein? Wir wissen genau, welche es nicht sein wird! Hat das etwa etwas mit der betreffenden Sprache zu tun? Bestimmt nicht. Die ist genauso gut wie jede andere auch. Und wie ist es mit Forth? Kommt denn keiner auf den Gedanken, daß es auch bei künstlichen Sprachen psychologische Barrieren geben kann? Ersetzen Sie "@" durch "FETCH" und "!" durch "STORE" - und schon ist ein Stück

Forth verlorengegangen. Versuchen Sie doch mal, Forth einzudeutschen. Das läßt sich an einem einzigen Abend erledigen. Aber das, was dabei herauskommt, ist das dann noch Forth? Und wenn schon wir selbst gewisse unbedeutende Äußerlichkeiten, die zur Einstufung von Forth als exotische Sprache beitragen, nicht ändern wollen, eben weil gerade die uns liebgeworden sind, wie können wir dann von anderen verlangen, daß die diese Äußerlichkeiten schlucken und mit wehenden Fahnen zu Forth überlaufen? Haben wir es überhaupt nötig, irgendwen von Forth zu überzeugen? Eine Glaubensgemeinschaft sind wir, das stimmt. Wir glauben fest an die Zukunft von Forth. Aber wozu eigentlich missionarischen Eifer zeigen? Die Blume blüht am besten im Verborgenen. Eine satte Blütenpracht kann erdrücken. Einen Wolkenkratzer am Potsdamer Platz wird man bewundern. In der Wolkenkratzer Schlucht der Fifth Avenue kommt man sich verlassen vor. Lassen wir doch Forth einfach im Verborgenen, in der Nische, in der es wächst, blüht und gedeiht, unangetastet von allen Ungläubigen!

Ist Forth noch Forth, wenn es die "Professionellen" in den Griff bekommen?

Lassen wir doch einfach alles so, wie es ist! Wer höhnen will, der höhne. Wir aber gehen hin und machen Dinge, an die andere nicht einmal zu denken wagen. In welcher anderen Sprache als Forth hätte ich es denn unternommen, ein Transputer-Betriebs- und -Programmiersystem auf die Beine zu stellen, das mir die volle Herrschaft über Speicher-verwaltung und Programmierung von sechs Transputerkarten verleiht? Ganz ohne OCCAM-Compiler. Von der Pike an. (Wortwörtlich von PEEK - und POKE - an.) Ohne irgendeine "geniale", aber umso kostenspieligere "Entwicklungsumgebung", bei der man nicht weiß, "wer" wo was zu welchem Zeitpunkt macht. Ein paar Jahre hat es zwar gedauert. Man ist ja kein "Programmierer". Aber Forth sei Dank war es zu schaffen.

Bei welchem Compiler wird einem denn gesagt, wer was wie und wo macht? Und warum so und nicht anders.

Beim GNU-C++-Projekt vielleicht. Sonst nicht. Compilieren ist aber nichts weiter als maschinelle Übersetzung. Wozu also die Geheimniskrämerei? Als ob wir im Mittelalter lebten, wo jeder neu entdeckte Lehrsatz per Kurier weitergereicht und gehütet wird! Was würde denn passieren, wenn man den Softwarehäusern verbieten wollte, das Newton-Verfahren anzuwenden? Wo würden die ihre Wurzeln hernehmen? Allgemeingut? Eben! Bei Programmier-techniken aber auch. Forth hat mir von Anfang an nichts verheimlicht. Natürlich muß man einsteigen, um wissend zu werden. Aber durch Quelltext-Hinterziehung zwangsweise ferngehalten wird man nicht. Und das ist eine der Stärken von Forth - die mit Forth selbst eigentlich gar nichts zu tun hat: Man kommt an die Quelltexte ran. Achten wir darauf, daß das so bleibt!



Zehn-Punkte-Programm

Schon von alters her, aber auch in neuerer Zeit sind 10-Punkte-Programme in den Vordergrund gerückt. Hier sind meine 10 Punkte:

1 Laßt auch den Hackern und Bastlern ihre Chance! Was ist denn an einem Hacker so verwerflich? Es lebe die Kreativität, auch wenn sie nicht von jedem gleich auf akademischer Ebene eingesetzt werden kann! Forth ist eine willkommene Nische. Alles geht, nichts ist unmöglich. Laßt uns diese Nische!

Lange Zeit hat man über BASIC und Spaghetti-Code die Nase gerümpft. Obwohl niemand in BASIC gezwungen war, Spaghetti zu produzieren. Dann hat man Sprachen eingerichtet, die Spaghetti-Code unmöglich machen. In Assembler nehmen sich verflochtene CMP-Abfragen recht vernünftig aus. In Forth, das dem Spaghetti an sich von Hause aus abhold ist, wird nichts direkt verboten. Ein paar vorzeitige EXITS oder LEAVES, und schon habe ich den besten Spaghetti -wenn ich es denn unbedingt will. Laßt mir diese Freiheit! Schimpft mich ruhig einen Hacker!

2 Sorgt dafür, daß brauchbare Bücher auch woanders gelesen werden! In unserer Zeit ist das ganz einfach dadurch zu erledigen, daß man beispiels-

weise den Franzis-Verlag dazu überredet, die beiden Standard-Bücher von Zech - natürlich ANS-erneuert - englisch herauszubringen. Unsere Buchhandlungen sind voll von Büchern amerikanischer Verlage. Und einige davon wurden und werden ja auch in unsere Sprache übertragen.

3 Sorgt dafür, daß die diversen Forth-Zeitschriften überall zur Verfügung stehen! Alle möglichen Zeitschriften, wie unbedeutend auch immer, sind in der Bibliothek der Technischen Universität München erhältlich, die Tagungsbände der internationalen Forth-Treffen nicht. Kaum daß man Dr. Dobb's Journal oder die ACM-SIG-Forth-Erscheinungen auftreiben kann. Von der Forth Dimensions ganz zu schweigen. Man überlege sich das einmal. In der Forth Dimensions wird ja Forth überhaupt erst "gemacht".

4 Laßt auch den Liebhabern ihre Chance! L'art pour l'art. Nur so aus Spaß. Denkt daran, daß nicht jeder gleich eine Ausbildung als Ingenieur mitbringt, in einem Entwicklungslabor sitzt und sprechende Waschmaschinen

konstruieren möchte. Maler setzen Bildelemente zu einem "nutzlosen" Ganzen zusammen. Schachspieler üben sich in strategischem Denken. Mir macht es Spaß, in Forth Dinge anzugehen, die eigentlich nur "Experten" vorbehalten sein sollten. Man kann doch nicht in jedem Fach gleich Experte sein. Soll man deswegen auf berufsfremde oder manchmal sogar berufsähnliche Betätigung verzichten? Mit der Chirurgie als Hobby ginge das natürlich nicht. Aber bei Forth doch wohl ohne weiteres.

5 Treibt die Standardisierung von Forth nicht zu weit! Jede Standardisierung ist eine Beschneidung. Forth ist keine Sprache, nicht einmal nur eine Programmierphilosophie. Forth ist eine Idee. Forth ist ein offenes System, bei dem Sprache und Werkzeug zur Bearbeitung dieser Sprache ineinander übergehen. Es gibt inzwischen andere Vorschläge, bei denen Sprache, Compiler und Interpreter ein Ganzes bilden. Die sind überflüssig. In Forth ist schon alles enthalten. Ständig hören wir, "in C++ kann man dieses nicht, in Turbo-PASCAL jenes nicht, in FORTRAN das

ganz andere nicht, in BASIC überhaupt nichts nicht." In Forth geht alles. Alle erdenklichen Programmier-Paradigmen sind schon eingebaut. Laßt es dabei! Laßt es nie soweit kommen, daß jemand sagt, in Forth gehe das oder jenes nicht!

6 Fangt nicht an, auch in Forth Informationsverheimlichung zu betreiben! Ein Anwendungsprogramm ohne Header, mit nur den nötigsten Worten in ein "Kompaktum" gegossen, ist eine schöne Sache - für den Anwender, den Endbenutzer der sprechenden Waschmaschine. Als Herrscher im Reich meiner Forth-Systeme bin ich nur bedingt daran interessiert: Wir sind es ein für allemal leid, mit Textverarbeitungssystemen in EXE-Form ohne Quelltext umgehen zu müssen. Wie leicht wäre es beim Turbo-PASCAL-Editor 3.0 (mein über Jahre hinweg bevorzugtes "Textverarbeitungssystem") gewesen, einige Dinge nach eigenem Geschmack einzurichten, wenn man nur den Quelltext gehabt hätte. Dazu braucht man nicht groß programmieren zu können. Beim Gerdes-Tastaturtreiber in Assembler habe ich das ja auch geschafft. Learning by doing. Natürlich kann man beispielsweise WinWord über WordBasic ummodellern. Aber wozu dann die immer teureren neueren Versionen? Es scheint eben doch nicht alles zu gehen. Und der Quelltext, auch von früheren Versionen, wird uns auf Ewigkeit vorenthalten. Nicht so bei Forth. Laßt es dabei! Laßt Forth so zugänglich wie bisher!

7 Sorgt dafür, daß die Computer-Fachzeitschriften von dem unseligen Standpunkt herunterkommen, es interessiere sich sowieso niemand für Forth! Das ist einfach nicht wahr. Hätte ich vor zwei Jahren, nach Anschaffung eines Laser-Druckers, in der c't einen Forth-Artikel über die Umsetzung von NEC-P60-Sequenzen in HP-Sequenzen gefunden, hätte ich die betreffende Ausgabe (statt sie nur im Supermarkt zu lesen) sofort gekauft. Welches Forth, wäre mir dann egal gewesen. Die Anpassungsarbeit hätte mir Spaß gemacht.

DOS-International hat 1987 SForth auf die Beine gestellt und sich später für NAXOS von Klaus Peper, das auf SForth aufbaut, verwendet (in Turbo-PASCAL programmiert - man müßte

wirklich mal Turbo-PASCAL in Forth programmieren). Kein einziges "Kilopaket" aber, kein einziger "Tip oder Trick" wurde jemals in NAXOS angeboten. Man muß das Interesse auch wecken. Das geht nicht, wenn sich die Zeitschriften querstellen.

Richtig, COBOL, ADA, PILOT werden auch nicht behandelt. Forth ist aber etwas ganz Besonderes. Verdient es Forth wirklich, durch Nichtbeachtung gestraft zu werden?

8 Ruft Forth-Projekte ins Leben, an denen sich alle beteiligen können! Jeder, der kann, liefert einen Baustein. Das einigt, reizt und spornt an. Linux wurde in einem halben Jahr hochgezogen. Weltweit. Alles in C, mit Assembler da, wo nötig - und mit allen Quelltexten. Wieso geht denn sowas nicht in Forth? Linux wurde von einem "Verrückten" begonnen - und von Verrückten weitergeführt. Gut, aber so verrückt und besessen wie die sind wir doch in Forth schon lange (?) Linux ist mir zu bombastisch. Ich brauche keinen Superuser mit Sonderzugriffsrechten auf Ebene 1 bis 5. Ich bin mein eigener Superuser. Ich möchte mit meiner Maschine nach eigenem Gutdünken umspringen können. Ein schönes 32-Bit-DOS, bei dem ich meine 32 Megabyte Speicher nach Herzenslust linear selbst aufteilen kann. Ohne übertriebene Schutzmechanismen. (Code darf sich selbst nicht verändern - wo gibt's denn sowas?) Und das alles in Forth. Man könnte es DORTH oder FOS nennen. Ganz unten angesetzt. BIOS und noch viel tiefer. Wie meinen ? SUN ? Ja, natürlich. Ich meine aber, zum Nulltarif. Und mit allen Quelltexten. Selbstdisziplin? Die haben wir in Forth gelernt.

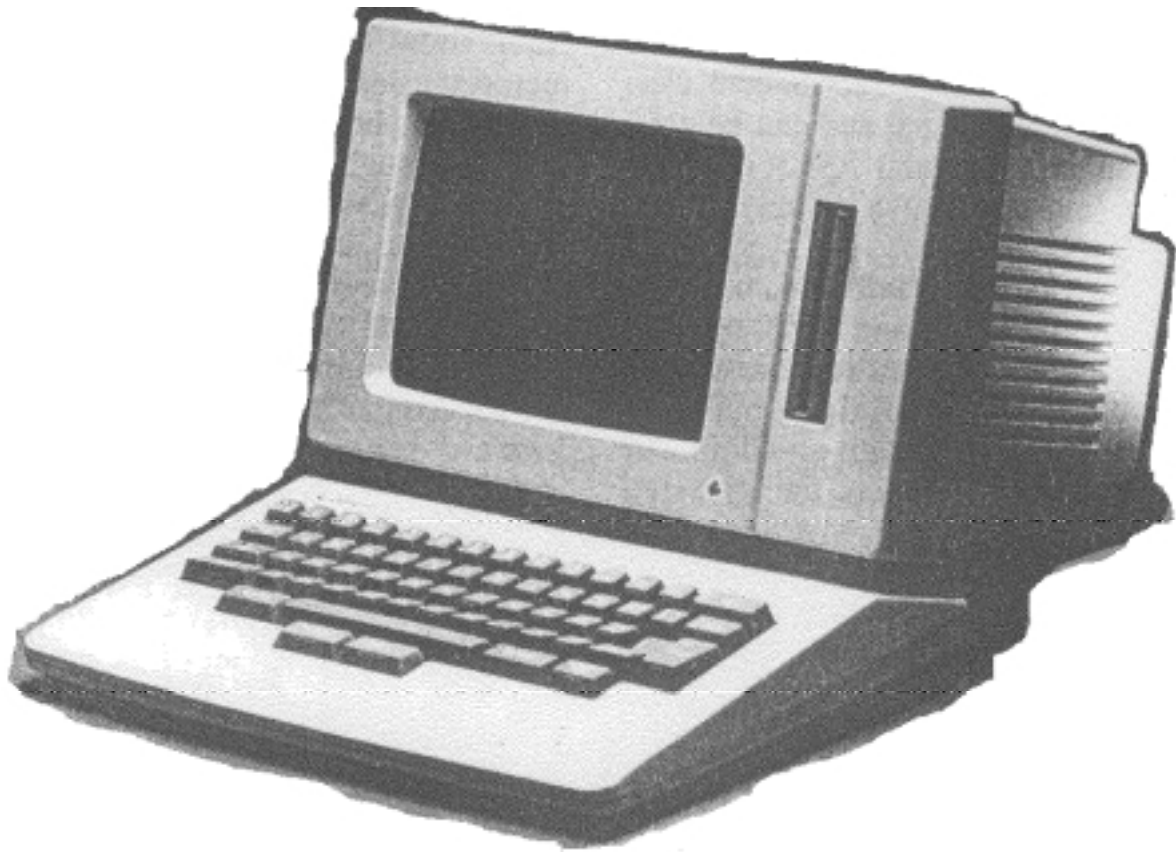
9 Warum nicht mal ein superaktuelles Forth-Handbuch in Gemeinschaftsarbeit schreiben? Jeder, der es sich zutraut, einen Abschnitt. Und das vielleicht sogar in englischer Sprache. Ich bin sehr gegen die Vorherrschaft des Englischen. Je besser man es kann, desto stärker merkt man, wie sehr man doch im Nachteil ist. Man kann nie das ausdrücken, was man eigentlich will. Aber mathematische Bücher werden beispielsweise im Springer-Verlag schon seit Jahren nur noch englisch

veröffentlicht. Und wenn Forth auf eine "professionelle" Ebene, was auch immer das bedeutet, gehoben werden will, dann geht das sicher nur über Englisch. Wir müssen uns eben unterordnen. Natürlich verlieren wir nichts dabei. Jeder auch nur einigermaßen Interessierte in Deutschland spricht Englisch. (Wie es umgekehrt in der Welt selbst mit dem bloßen Verstehen unserer eigenen Sprache aussieht, ist eine andere Frage.)

10 Schafft Sammlungen von mathematischen Grundalgorithmen in Forth - und Ihr erzwingt die Aufmerksamkeit der wissenschaftlichen Anwender! Die Frage, ob Forth eine Sprache ist oder nicht, wird dann zweitrangig. Es gibt C++-Bibliotheken, es gibt "Mathematical Recipes in FORTRAN". Seit über zehn Jahren laufen die Diplomanden nicht mehr mit lochkartengestanztem ALGOL ins Rechenzentrum, sondern setzen sich an einen der vielen institutseigenen PCs und wälzen Turbo-PASCAL-Handbücher. Das hat alles seine Gründe. "Mathematical Recipes in Forth" wäre doch eine schöne Sache? Natürlich Code-Definitionen, mit dem eigentlichen Inhalt in Maschinensprache. Aber Forth als Verwaltungsrahmen. Ohne Präprozessor, Parser, Linker und was weiß ich noch was. Forth kann das ja bestens. Nehmt doch einfach mal den Knuth (einen davon) und übersetzt ihn ins Forthische!

Fred Behringer promovierte 1969, habilitierte 1975 und lehrt seit Jahren "Operations Research" an der Technischen Universität München. Er kann sich auf englisch, holländisch, italienisch und französisch verständigen. Seine Lieblingssprache aber ist Forth. Eine ähnliche Sprache, zur digitalen Wartung von Analogrechenanlagen, hat er 1970 entwickelt. Seit dem Aufkommen der preisgünstigen Heimcomputer, seit etwa 1981, fasziniert ihn Forth als Sprache, die viele seiner ursprünglichen Träume wahr werden läßt, ohne daß er sich gleich der Informatik verschreiben müßte. Informatik hat es in Freds besten Jahren noch gar nicht gegeben. Fred stand schon 1983 mit unserem jetzigen Redakteur, Claus Vogt, in Briefkontakt. Damals ging es um Forth auf dem C16 und anderen Commodore-Maschinen. "Die Zeiten haben sich total geändert", sagt Fred, "Forth aber hat für mich nichts an Faszination verloren."

□



Jeff Raskins "Canon Cat"

von Rafael Deliano

Steinbergstr. 37, D-82110 Germering, 089/841 83 17

Ein bemerkenswerter Tischcomputer. Bekannt wurde er durch den Mann, der ihn realisierte und sein radikales Konzept. Die Software wurde in FORTH implementiert und der Benutzer hatte auch direkten Zugriff auf den FORTH-Interpreter.

Stichworte: Canon_Cat Raskin Macintosh

Jeff Raskin ist ein Mann mit vielen Interessen. Ursprünglich an der University of California (UCSD) als Professor für Kunst und Computer zuständig, hinderte ihn das nicht, sich tiefgreifend mit Elektronik und Modellflugzeugen zu beschäftigen. Nachdem er von der Hochschule genug hatte, sattelte er auf Musiker um. Mitte der 70er Jahre kamen die Mikroprozessoren. Er sah die Möglichkeiten und gründete eine Firma, die technische Dokumentation für diesen Bereich erstellte. So kam es, daß er 1978 als Manager of Publications der 31. Mitarbeiter der Firma Apple wurde. Als Manager of Advanced Systems begann er 1980 mit drei Mitarbeitern am

Konzept des Tischcomputers der nächsten Generation zu arbeiten. Projektname: Macintosh.

Die Arbeit am Mac war eher Raskins privates Steckenpferd als der Hoffnungsträger der Firma. Nachdem der Nachfolger des Apple II, der Apple III zum Flop geworden war, konzentrierte sich Apple damals voll auf Lisa, die sich direkt an den Altos von Xerox PARC anlehnte.

Macintosh

Auch Raskin hatte während seiner Hochschultätigkeit Xerox PARC besucht. Er war beeindruckt, sicherlich auch inspiriert. Sein Ur-Mac war trotzdem anders. Er prägte für sein Konzept den Begriff "Desktop Appliance". Der Computer als Haushaltsgerät. Haushaltsgeräte sind billig und einfach zu bedienen. Preis und einfache Bedienbarkeit sollten dem Kleincomputer den Durchbruch auf den Massenmarkt der Konsumgüterindustrie bringen. Die von der Geschäftsleitung angeforderte Projektstudie "Annie" für \$500 Endverbraucherpreis erwies sich als nicht machbar. Für den Macintosh mußten deshalb \$1000 angenommen werden. Das bedeutete Herstellkosten von \$300. Und auch das reichte nur für die simple Hardware eines 8-Bit Homecomputers knapp oberhalb des Apple II. Deshalb war Raskin als CPU auf den 8-Bit 6809 und damit einen 64k-Adreßbereich fixiert und lehnte den 16-Bit 68000 der Lisa als zu teuer ab. Auch er wollte bit-mapped Grafik, beschränkte sie aber ursprünglich auf 256x256 Pixel. Schon die ersten Prototypen mußten auf 384x256 Pixel erweitert werden, damit

The Toaster



Jeff Raskin: "Let's consider what it would be like if a computer company had to design a toaster. You would wake up and you want a piece of toast for breakfast. The first thing you do is to switch the toaster on. If it was designed by General Electric, you'd put the toast in and off you'd go, but no, this toaster was designed by a computer company. So what happens? First of all it does a two minute toast check. Then you put in the system disk and boot the system. After that you put in your breakfast disk and then you type "Load TOASTED.CODE". So, what happens next? Up comes the menu. It asks, "What kind of bread are you going to have?" If it is a California program, it'll say croissant, bagel, English muffin, whole wheat, and at the bottom, of course, white bread. They're labeled A, B, C, D, E, so you hit C because you feel like a muffin this morning. Nothing happens, because you forgot to hit return, You'd think the machine would be smart enough to respond to C, but you have to hit return anyway. Do you think it does anything now? Of course not. It's designed by a computer company. It says "Are you sure?" Now you are ready to throw it through the wall. Are you mad yet? Haven't you be mad at computers for years? But because you spent a couple of thousand dollars on it, you put up with all this stupidity, and so does the rest of the world. Millions of people go through nonsense like this every time they use a computer. So you type "Yes" and hit return, but get an error message because you were supposed to hit something else. You consult the manual, but it tells you nothing because the people who wrote it were describing a prototype system that has changed. Finally you put the bread in slot two, indicate whether you want light, tan, seriously burnt bread and the computer asks, "Do you want to save this breakfast so you don't have to redo this again?" So you type "Yes," and it tells you to put a disk in slot one, but you discover you are out of formatted disks. You call up the dealer and ask if there's any way to save this while formatting the disk without losing everything you've already done this morning? He says, "Yeah, just buy this \$3,000 hard-disk system with MS-DOS 9.8 and that will solve all your problems. It comes with a manual and a handtruck." The handtruck is for moving the manual. And you are already late for work..."

Aus: Susan Lammers "Programmers at Work" Microsoft Press 1986 ISBN 0-914845-71-3

□

zumindest rudimentäre Textverarbeitung machbar wurde. Für eine Maschine mit 64k-Speicher war das die Grenze. Der endgültige Mac von Apple hatte 512x256 Pixel, was büromäßige Textverarbeitung mit 80 Zeichen/Zeile ermöglichte.

Die Maus der Lisa hielt Raskin für eine Spielerei. Sein Mac war für nützliche Anwendungen wie Textverarbeitung optimiert. Und dafür genügt eine Tastatur. Er war durchaus bereit, das visuelle Interface von Xerox PARC zu übernehmen. Aber die Steuerung hatte von der Tastatur aus zu erfolgen, nicht durch eine Maus. Und wenn überhaupt eine Maus, dann nicht mit 3 Knöpfen wie bei den frühen Lisa-Prototypen, sondern bestenfalls mit einem Knopf. Zumindest in diesem Punkt hat er sich bei Apple durchgesetzt.

Während der Apple][noch aus Grundgerät, Monitor und Floppys bestand, die durch einen Kabelsalat verbunden waren, war der Mac kompakt und sogar etwas portabel. Alle Baugruppen waren in einem Gehäuse untergebracht. Auch die Tastatur war starr mit dem Gehäuse verbunden.

Nach dem Einschalten sollte sofortiges Arbeiten möglich sein. Betriebssystem und wesentliche Applikationen wie Textverarbeitung waren deshalb im ROM enthalten. ROM ist als Programmspeicher unflexibel, aber billig. Für Software von Fremdentwicklern hatte dieses Konzept keinen Platz. Die Floppy diente zum Transfer von Daten, nicht von Software. Das Gerät hatte selbstverständlich keine Slots. Das hätte Mehrkosten und Komplexität für den Anwender bedeutet.

Der Ur-Mac kam nicht über das Prototypenstadium hinaus. Nachdem Firmengründer Steve Jobs bei Apple aus allen wichtigen Positionen verdrängt worden war, übernahm er die kleine Entwicklungsgruppe und setzte Raskin vor die Tür. Es folgte eine langsame Rückentwicklung zu einem konventionelleren Kleincomputer. Die von Jobs durchgesetzte grafische Benutzeroberfläche mit Maus erreichte zwar die einfache Bedienbarkeit, die das "Haushaltsgerät" auch forderte. Der damit verbundene Aufwand an Rechenleistung und Speicherverbrauch zerstörte jedoch die Kostenstruktur. Der nunmehr



angestrebte Preis von \$2000 mußte bei Markteinführung nochmal auf \$2500 erhöht werden. Der Macintosh wurde zum Yuppie-Spielzeug, nicht zum Model T der Massen. Viele der Eigenschaften, die von Raskins Prototypen beibehalten wurden, z.B. der Mangel an Slots, erwiesen sich für eine konventionelle Maschine am Markt als Handicap. Insgesamt gesehen gilt Raskin heute vielleicht zu Unrecht als geistiger Vater des Macintosh. Sein Macintosh war ein anderes Gerät.

Information Appliance

Nach seinem Ausscheiden bei Apple 1982 gründete Raskin die Firma "Information Appliance" und verfolgte sein Konzept weiter. Einige Jahre lang mußte er sich mit Kleinkram zufriedengeben, z.B. der Entwicklung einer Apple]-Einschubkarte für Textverarbeitung. Der Erfolg des Macintosh tat seiner Reputation jedoch gut. Die japanische Firma Canon ist für Drucker und Kopierer ein führender Anbieter. Sie hatte schon früher erfolglos versucht, im sich abzeichnenden Kernbereich der Bürogeräte, dem Kleincomputerbereich Fuß zu fassen. In ihr hatte Raskin den nötigen Geldgeber gefunden der auch über Fertigung und Vertrieb verfügte.

1987 ist das Textverarbeitungssystem "Canon Cat" fertig. Die Hardware entspricht etwa dem, was man in Europa von einem BTX-Terminal gewohnt ist. Tatsächlich ist auch ein 1200-Baud-Modem eingebaut. Die Tastatur ist starr mit dem Gehäuse verbunden, der Bildschirm klein und schwarz/weiß. Zusätzlich ein 3,5"-Floppy, fertig ist der Kleincomputer. Die Software war weniger frugal. Sie sollte anwenderfreundlich sein und das braucht Rechenleistung. 256kByte Firmware im ROM werden von einem 5MHz-68000 angetrieben. Die 256kByte RAM reichen für etwa 80 Seiten Text. Die Diskette wird dabei immer komplett ins RAM kopiert um maximale Arbeitsgeschwindigkeit zu erreichen. In weiteren 128kByte ROM ist ein 90k Worte umfassendes Wörterbuch der amerikanischen Sprache untergebracht. Der Bildschirm verwendet kaum Grafik. Das Gerät lehnt sich in der Handhabung so weit wie möglich an eine konventionelle Schreibmaschine an.

Raskin ist der Überzeugung, damit dem typischen Anwender, also der Sekretärin, den einfachsten Übergang auf einen Computer zu ermöglichen. Bei Problemen kann man sich umfangreiche kontextsensitive Hilfsmeldungen auf den Bildschirm holen.

Was viel zur Zufriedenheit des Benutzers beiträgt, ist Geschwindigkeit. Alle Funktionen werden mit minimaler Verzögerung ausgeführt. Selbst der Diskettenzugriff ist, wo er nicht vermieden werden kann, sorgfältig optimiert worden.

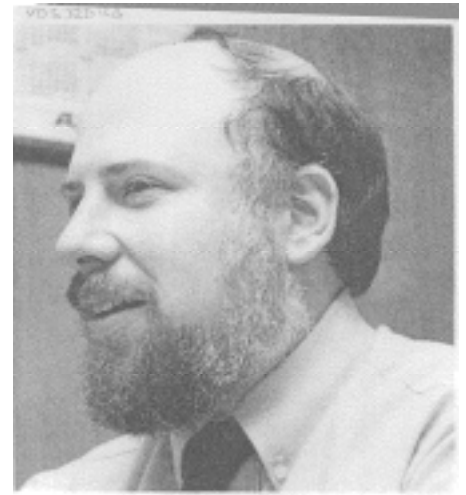
Normalerweise markiert man Textabschnitte mit Inversdarstellung, um sie mit den Copy&Paste-Funktion der Textverarbeitung bearbeiten zu können. Das ist bei der Cat genauso. Handelt es sich jedoch um eine mathematische Formel, wird von der Cat das Ergebnis berechnet. Beim nächsten Tastendruck erscheint wieder die ursprüngliche Formel. So wurde ein Taschenrechner integriert.

Handelt es sich beim markierten Text um ein FORTH- oder 68000-Assemblerprogramm, wird es ausgeführt. Das System wurde in FORTH implementiert. Der Anwender hat vollen Zugriff, was die Entwicklung eigener Programme vereinfacht.

FORTH

Raskin ist kein FORTH-Enthusiast. Allerdings sind seine Neigungen natürlich unkonventionell. Er hatte bei seinen Computerkursen an der UCSD die selbstentwickelte, sehr einfache flow-language verwendet. Für den Ur-Mac war die noch zu entwickelnde Implementierungssparche "APL", Apple Programming Language vorgesehen. Seine engsten Mitarbeiter hatten genau wie er akademische Neigung und waren für die Einführung von Pascal bei Apple verantwortlich. Der endgültige Mac wurde ein Assembler und Pascal programmiert. Das heutige Mac-OS ist natürlich inzwischen auf C umgeschrieben. Vermutlich wäre "APL" eine Pascal-Variante für Systemprogrammierung geworden.

Für Information Appliance war die Wahl jedoch wieder offen. Raskin: "I decided to implement it in FORTH, because FORTH is a rather compact language and is inexpensive to imple-



ment. It's not my favorite language, but I thought it was suitable for this particular application. I always believe you should use the right tool for the job."

Finis

Gorbatschow hatte auch hier wieder recht: wer zu spät kommt.... Ein derartiges Gerät wäre 1983 eine Durchbruch gewesen. 1987 waren die Features einfach nicht mehr konkurrenzfähig. Die Maschine war mit \$1500 außerdem zu teuer. IBM-Clones boten damals schon mehr fürs Geld. Es wurden nur geringe Stückzahlen verkauft. Canon hat in den USA nicht die Möglichkeiten für Marketing wie Apple. Vertrieb in Japan war für ein ASCII-orientiertes Gerät problematisch. Ihre wenigen Benutzer z.B. auch Byte-Redakteure schienen sie gemocht zu haben. Auch Information Appliance existiert nicht mehr. Raskin selbst ist jedoch munter. Kürzlich erschien von ihm in IEEE "Spectrum" die Rezension eines Buchs über NeXT. Er ist kampflustig wie immer und seine Meinung über Steve Jobs hat sich auch nicht geändert.

Literatur:

Produktvorstellung in "Byte" 10/1987 S.121

Interview mit Raskin in: Lammers "Programmers at Work"

Interview mit Raskin in: "Byte" 8/84 S.347

J.S. Young "Steve Jobs The Journey is the reward"





Forth Online



Olaf Stoyke

os@cs.tu-berlin.de

Ab dieser Ausgabe enthält das Forth Magazin eine neue Kolumne. Olaf Stoyke hat sich freundlicherweise bereiterklärt, uns über FTP, WWW, das Internet und all jene neuen Medien auf dem Laufenden zu halten, die wir normalen Zeitungsleser nur als kryptische Abkürzungen aus der Zeitung kennen.

Hinweise auf forthrelevant Verdrahtetes nimmt er gerne entgegen. Schließlich kostet auch ein Globetrotter in der Welt der weitreisenden Bits stets gerne die Kochrezepte anderer Digital-Gourmets. (red)

Es gibt eine recht neue Seite im WWW (oder Web), was an und für sich nicht so weltbewegend ist, da dieser Dienst inzwischen allumfassend in fast jedermanns Rechner und damit (bald - laut Bill Gates) in jedermanns Heim kriecht. Diese Seite nun enthält recht übersichtlich aufgelistet Artikel aus der Newsgroup *comp.compilers* seit 1986.

Auch das ist nicht so neu, da diese Artikel immer irgendwo abrufbar oder als Archiv kopierbar sind. Für den deutschen User ist diese Seite deshalb so interessant, weil einerseits die Informationen klar gegliedert sind und weil zusätzlich das ganze noch aus der Nachbarschaft kommt, aus der Schweiz nämlich:

<http://cuiwww.unige.ch/OSG/Vitek/Compilers/index.html>

Schon so manche gute Idee wurde in *comp.compilers* diskutiert und produziert und da hier keine langen Wartezeiten auftreten, gehört diese Seite eigentlich in das heimische Bücherregal... oder halt in die Bookmark-Liste von Netscape.

Unter <http://www.x86.org/> (Schön kurze Adresse - Claus wird sich freuen) finden sich all jene Informationen, die - nach Angaben des Erstellers - „Intel nicht verraten wollte“. Nun ja, dachte ich mir, so doll kann das ja nicht sein, denn entweder findet man dort (wieder) 1001 Artikel über den FDIV-Bug im Pentium-Prozessor oder 2002 Vermutungen und Vorabberichte über den P6 (wird er wohl 'Hexium' heißen? Nein, Pentium Pro ist der Name!). Die Top-Level-Verweise lauten: "Undocumented Processor Information", "Books and Magazine Articles", "Productivity Enhancements", "MR BIOS", was auf einen BIOS-Hersteller deutet, und wie erwartet: P6 Information".

Da ich mal voraussetze, daß harte Fakten willkommen sind, habe ich mir einmal die relevanten Teile angesehen und wollte, neugierig wie ich nun einmal bin, sehen, was ich über den Pentium noch nicht weiß und wie ich produktiver arbeiten kann... Unter "Undocumented Processor Information" findet

man Beschreibungen von offiziell nicht existierenden Befehlen, Möglichkeiten von sonst 'verbotenen' Zugriffen auf interne CPU-Register (TR4, TR5, DR7), etc. Interessant ist auch das Shareware-Paket 4P Version 3.1.3: Es enthält alles, was es zu den Intel-CPU's an Informationen gibt und zusätzlich ein Programm, das die CPU des eigenen Rechners auslotet; Motto: „Alles, was Sie schon immer über Ihre CPU wissen wollten, aber nie zu fragen wagten“.

Unter <http://users1.ee.net/brey/> findet sich die Seite des Fachbuchautors und Professors Barry B. Brey, der ein gutes Dutzend Bücher über Prozessoren und deren Programmierung in Assembler geschrieben hat (die übrigens alle auch dort erwähnt sind...). Interessant wird es nun unter dem Punkt "other links", wo tatsächlich und völlig überraschend nicht etwa öde Listen von Web-Links auf das genervte Auge lauern, sondern wohlthuend gut gestaltet Verweise auf alle möglichen und vor allem an einem Ort gesammelten Server der verschiedenen Hersteller und Anbieter residieren: Die Rubriken sind: "Video Drivers", "Printer Drivers", "Modems", "Computer Company Support Sites", "Software Links", "Hardware Links", "Web Browsers and Editors", "Graphics for your webpage", ... Wozu noch alle Web-Adressen speichern, wenn es Seiten wie diese gibt?

Nach den recht allgemeinen und den zu hardware-spezifischen Verweisen nun ein reines Forth-Archiv auf Seite <http://www.paisley.ac.uk/~cis/forth/index.html>, The Forth Page genannt. Unter "Introduction to Forth" findet man einen HTML-Artikel, der "Forth underview" betitelt ist. Für einen Einsteiger (wie mich) genau die richtige Lektüre, um die Interna eines Forthsystems näher kennenzulernen. Weitere Stichpunkte jener Seite: Eine Literaturliste, die man interaktiv abgrasen kann, eine Liste von Anbietern kommerzieller Forth-Lösungen sowie eine Seite, die zum Rest der Forth-Welt verzweigt...

"FORTH on the WEB" nennt sich eine Web-Seite, die alle möglichen Informationen rund um die Sprache und ihre Anwendungsgebiete zusammenzubringen sucht: Die Links zu diversen FORTH-Archiven, eine Liste häufig gestellter Fragen (FAQ), Einführungen in die Sprache. Der letzte Punkt läßt vermuten, daß man auch dem Einsteiger ermöglichen will, mit diesem Angebot die ersten Schritte zu versuchen und sich mit dieser Sprache anzufreunden. Das Problem hierbei ist jedoch, daß die zahlreichen Verweise auf andere Rechner nicht kommentiert sind und beim Anklicken eines solchen Links nur mehr oder weniger übersichtliche und vor allem unkommentierte Inhaltsverzeichnisse auf dem Bildschirm zu bewundern sind.

Allein die wohl grenz- und rechnerübergreifende Popularität des FPC-Systems hat wohl dazu geführt, daß es zum einen namentlich erwähnt wird und zum anderen einen Verweis auf ein sehr detailliertes und gut zu überblickendes Angebot von Fred Warren auf <http://www.efn.org/~fwarren/fpc.htm> gibt. Ansonsten gilt natürlich wie immer: Anschauen, Ausplündern und bei Gefallen weitererzählen. Ach übrigens, FORTH on the WEB ist unter <http://pisa.rockefeller.edu:8080/FORTH/> zu erreichen...

□

Datum und Uhrzeit des Compilierens
festschreiben

Timestamping a file

von Jörg Staben
Hagelkreuzstr. 23; D-40721 Hilden
Tel.: 02103 / 24 06 09

Im Netz gab's unlängst eine interessante kleine Diskussion zwischen Wolfgang Allinger, Mark A. Flacy (flacy@bnr.ca), Tom Zimmer (ZForth@ix.netcom.com) und Jörg Staben.

Stichworte: TCOM time Versionskontrolle

W. Allinger hatte ein schmerzliches Problem mit F-PC & TCOM: Er verwendet für seine Programmentwicklung unter F-PC und anderen Forth-Systemen Wörter, die das Datum und die Zeit des Compilierens in 2CONSTANTen ablegen.

Dies ist Allingers Meinung nach der einfachste Weg zu einer Versionskontrolle für Programme, wenn EPROMS für embedded controller hergestellt werden sollen. Man erpart sich das Klebeschildchen auf dem Baustein, hat ihn aber dennoch eindeutig gekennzeichnet. Das Problem lag nun darin, daß TCOM diese beiden Zeilen:

```
getdate 2CONSTANT $date
gettime 2CONSTANT $time
```

mit dieser Fehlermeldung quittierte:

```
*ERROR* STAMP.SEQ(line#...)[ 2CONSTANT ] Not enough Parameters
```

Die Vermutung, daß TCOM sowohl GETDATE als auch GETTIME nicht richtig interpretiert, sondern nur einen 16Bit-Wert auf dem Stack zurückließ (warum eigentlich?), ließ sich durch

```
0 GETDATE 2CONSTANT $$DATE
```

beweisen. Was allerdings immer noch nicht das gewünschte Datum als Ergebnis lieferte.

M.Flacy vermutete zuerst, man könne interpretiertes F-PC nicht mit compiliertem TCOM mischen. Dies ist so nicht richtig, wie dieser Text als - dank Tom Zimmer's Hinweisen - lauffähige Programm-Version zeigt.

```
\fpc anew *timestamp*

\fpc defined TARGET nip not \ FPC kennt TARGET nicht
\fpc #IF Vocabulary TARGET #THEN
  FORTH getdate TARGET 2Constant $date
  FORTH gettime TARGET 2Constant $time

: .stamp ( -- )
  cr ." Program compiled at: "
  $date form-date count type 2 spaces
  $time form-time count type 2 spaces ;
\fpc .stamp
```

TARGET - oder auch [TARGET] genannt - ist lediglich der Name eines Vokabulars. Das Wort FORTH oder [FORTH] schaltet auf das FORTH Vokabular um und erlaubt somit den Zugriff auf GETDATE/GETTIME. Dann schaltet [TARGET] wieder zurück auf das Vokabular des Target Compilers.

Damit wird die Target Version von 2CONSTANT verfügbar, das die zwei 16Bit-Werte vom Stack nimmt und sie im Target Wörterbuch and make them available in the target dictionary if the constants name is used in a target word.

Der Grund liegt darin, daß Konstanten nicht wirklich kompiliert werden, wenn sie nicht benutzt werden. Konstanten werden erst dann in das Zielprogramm kompiliert, wenn sie in einer Definition benutzt werden. Dann werden sie als "inline assembly instruction literal loads". So kompiliert 2CONSTANT einfach zwei "inline literal loads" und schiebt den ersten Wert vor dem zweiten auf den Stack.

Woher die Lösung OHNE Tom Zimmer?

Mark A. Flacy schreibt, nicht zu Unrecht, er habe eine ganze Zeit daran gesessen, um die Antwort auf diese Frage zu finden. Ist die Dokumentation des F-PC/TCOM doch unvollständig?

Tom Zimmer hat an anderer Stelle in "F-PC, TCOM and Win32Forth" ganz allgemein über die Philosophie von F-PC gesprochen; hier weist er in Kürze noch einmal auf das grundlegende Vorgehen hin:

Er benutzt einfache Werkzeuge, um sich an Code-Beispiele zu erinnern oder sich weitere Programmier Techniken anzuschauen. Das Werkzeug für dieses Problem war LOOK.COM, ein kleines TCOM Programm ähnlich dem (Unix-Befehl) GREP, der Dateien nach einem Wort oder einem Ausdruck durchsucht. Hier ging die Suche auf GETDATE, die fast sofort zu einem Beispiel-Programm des TCOM, VED, führte. Dort finden sich folgende zwei Zeilen:

```
create veddate [forth] getdate [target] , ,
create vedtime [forth] gettime [target] , ,
```

Variablen als Lösung?!

J. Staben schlug unterdessen die Lösung mit 2Variablen vor; diese hätte dann so ausgesehen:

```
2Variable $D 2Variable $T

: make_stamp getdate $D 2! gettime $T 2! ;

: sign-on
  make_stamp cr ." SIGN: "
  $D 2@ form-date count type 2 spaces
  $T 2@ form-time count type 2 spaces ;
\fpc sign-on \s
```

Leider wies W.Allinger sofort daraufhin, daß diese 'Lösung' keine ist: Denn 'make_stamp' wird ausgeführt, wenn 'sign-on' ausgeführt wird und gibt dann beim Programmstart nur das aktuelle Datum und die aktuelle Zeit aus, nicht Datum und Zeit des Kompilierens. Was sich übrigens im F-PC leicht zeigen ließ...

Die Entwicklung dieses Programmierproblems zeigt (wieder), wie faszinierend funktionell die Programmentwicklung in F-PC/TCOM ist. Ähnlich wie in einer integrierten Microsoft/Borland-Umgebung ala TurboC wird innerhalb des TCOM-Editors das Programm geschrieben, Sekunden kompiliert und ausgeführt. Zudem steht bei Bedarf, wie er hier vorhanden war, in Sekundenschnelle das F-PC als Interpreter zur Verfügung. Herrlich!!

Noch Platz !?!

auf der Fest-(tags-)Platte, im 16-bit Zahlenraum oder im Anfänger-Kopf?

von Arndt Klingelberg
Straßburger Str. 12; D-52477 Alsdorf

Im Zusammenhang mit dem Installationsprogramm für F-PC-ak landete ich mal wieder am Ende des 16-bit Universums: Alte F-PC-Nutzer und mittelalte tCOM-Installierer kennen diesen Bug: Festplattenpartitionen mit mehr als 64 MB ergaben mehr oder weniger Probleme. Auch einiges an älterer sonstiger Software läuft nur auf max. 64 MB Partitionen, teilweise sogar nur bis 32 MB. (Im letzten Fall wurden dann sogar nur 15-bit ausgenutzt oder anders gesagt: Vorzeichenbehaftete Rechen-Operatoren genutzt, siehe unten).

Stichworte: F-PC tCOM Compiler-Direktiven Zahlenformatierung DiskSpace

Mit System ...

Hochtrabend ausgedrückt handelt es sich nachfolgend um ein Stück 'System' Programmierung, aber um ein einfaches und gut für den Einstieg. Vor einer tCOM Endverarbeitung ist es sinnvoll, ersteinmal alles interaktiv mit vollem Komfort und allen DebugMöglichkeiten in F-PC testen. Also ist eine compatible CODE-Schreibweise gefordert.

Solche F-PC/tCOM-compatiblen Quelltexte sind einfach handzuhaben, sehen aber erstmal kompliziert aus. Der Trick besteht darin, die Quelltexte selektiv zu laden, das ermöglichen die CompilerDirektiven `·&&F` und `·&&T`. Sie heißen aus gutem Grund nicht wie im Original `·F` und `·T`. So müßte `·F` eigentlich vom Namen her bedeuten, daß für `F == F-PC` die Funktion `'\'` gilt, also AUSkommentiert wird! Wie immer bei doppelten Verneinungen ergibt das große MisTverständnis und daher bei F-PC-ak die C//UNIX entlehnte Schreibweise `'&&'` für: 'hier geht es weiter' bzw. `'AND'` (und `'||'` für das Gegenteil `'STOP wenn ...'` bzw. `'OR'`).

Die 'Platz'-Funktion, die wir suchen heißt 'Get Free Disk Space' und wird von DOS (unserem 'SYSTEM') als fixfertiger Service unter int \$21 function \$36 angeliefert. Wie dieser sogenannte (!) Interrupt zu bedienen ist, sagt uns z.B. so eine Bibel wie der 'Tischer' oder die ultimative Referenz von Schäpers (Buchbesprechungen siehe VD 3/93) oder Ray Duncan oder Im Register DL ist die Laufwerknummer und im Register AH die Funktionsnummer \$36 dem 'INT 21' anzuliefern. DOS spuckt dann auf Register AX die Anzahl der Sektoren je Cluster, auf BX die freien Cluster, auf CX die Bytes je Sector und auf DX die GesamtClusterZahl aus. Kann das 'Device' nicht gefunden werden, so enthält AX die höchstmögliche 16-bit Zahl \$FFFF == -1 == +65535.

Die einfachste Verfahrensweise in Forth bedeutet, ein Code-Wort zu schaffen, das Register-Orientierung in Stackorientierung wandelt, also die DriveNummer vom Stack zieht und die anderen vier Register dann auf den Stack legt. Als Luxusbonbon wandelt die Routine noch die mögliche \$FFFF

in AX zu einer Null. Damit liegt bei NICHT-vorhandenem Drive eine NULL oben auf dem Stack bzw. - wenn blind weiter gerechnet wird - wird nicht typisch 128 MB für Nichts ermittelt (die wundersame Drive-Vermehrung), sondern recht sinnvoll 0.0 !

Da ich zwar in Assembler für F-PC fit bin, aber für tCOM sich immer noch ab und zu Fragezeichen durch die Gehirnwindungen schlängeln, greife ich auf die HilfeTexte in F-PC-ak zurück. Dort (`·TCOM-<-->F-PC`) werden typische Beispiele und auch die wichtigsten Assembler Sequenzen in

File: dspace.seq vom 13.1.1994

```

1 \ \ DSPACE.seq  Is the disk full or going to get full ?          \ akg93nov19
2
3
4 \ akg94jan12  final revision for VD93Q4, just corrections of layout and NAME
5 \ modified criterium for not-enough-free-space beep
6
7
8 comment:  ~~~~~ \ akg93nov19
9
10 old filename: diskFULL.seq  but we are optimists, aren't we ?! arndt ! !!
11
12 compare:  ·sEDcode.seq  ·getDISKfree
13
14 ·sedit0.seq  ·?enoughDISK  ·?diskFULL  ·currentSIZE
15
16 TCOM :      there is a BUG sometimes with toom using <-> 25 lines !!! \ akg93dec02
17
18      ??dSPACE
19      ? Drive:      default: E:
20      ? free:       1 MB 1002 KB (= 2'074'624 Bytes )
21      ? total:     127 MB 184 KB (= 133'357'568 Bytes )
22      ? used:      125 MB 206 KB (= 131'282'944 Bytes )
23      ? per Cluster: 2048 Bytes
24      ??
25      ~~~~~
26 comment;
27
28 ·OPTIONS /OPT /DEFINITE          \ though a comment, read by TC0M !!!
29 {
30 &&F ONLY FORTH ALSO DEFINITIONS decimal
31 &&F && autoTitle          autoTitle          \ an automatic forget
32 code getdiskspace          \ return total and free space from disk

```

F-PC und tCOM gegenüber gestellt. Im Gegensatz zu F-PC hält tCOM TOS im Register BX (die oberste Stack Cell also intern im Register). Damit ähnelt tCOM dem Volksforth und auch dem ZEN-ANS-Forth. Return-Stack und Parameter-Stack sind vertauscht (siehe XCHG Befehl). tCOM ist subroutine-threaded, F-PC 'nur' direct-threaded. tCOM unne-
sted nicht über NEXT sondern geht mit RET aus einem CALL zurück. Verschiedene Makros ermöglichen es jedoch, den CODE recht einfach zwischen F-PC und tCOM kompatibel zu machen. Es werden hier Makros genutzt, da diese von tCOM typischerweise auch wieder einfach wegoptimiert werden können. So gibt es die Makros LOAD_BX und SAVE_BX, die sich aufheben können (durch Optimierung).

Im Source-Text habe ich bewußt die Code-Teile für F-PC oder tCOM und die gemeinsamen Teile in versetzten Spalten angeordnet.

16-bit als Qual(lität)

Nun ergibt sich das forthtypische Problem (ihr wenigen 32-bitler bitte weghören), daß bei weiteren sinnvollen Berechnungen sehr schnell der 16-bit Zahlenraum verlassen wird. Wird in kByte gerechnet, so sind das eben die berüchtigten 64 MB, in Byte gerechnet ergibt sich nur noch Quatsch. Doppelt genau multipliziert muß aber eigentlich nicht werden, die Startgrößen sind 16-bit, aber das Ergebnis beträgt dann 32-bit. Hier bieten sich die 'M'ixed Operatoren an. In der Vergangenheit gab es Vorlieben für diese oder jene Schreibweise. Der ANS-Norm haben wir es zu verdanken, hier wieder eine eindeutige Richtschnur vorzufinden. In der

F-PC original Version und tCOM sind U*D statt ANS UM* definiert, *D statt M*. Ein Alias ist aber jeweils vorhanden. Genauso wird 'M'ixed dividiert: aus einer doppelt genauen 32-bit Zahl wird durch Division mit einer 'normalen' 16-bit Zahl wieder eine 16-bit Zahl: UM/MOD bzw. M/MOD .

Das vorangestellte 'U' bedeutet jeweils 'U'nsigned, damit kann der volle und nicht nur der halbe (=positive) Zahlenbereich bei gleicher bit-Breite genutzt werden.

Und warum rechnet man nun mit solch komplizierten Operatoren statt direkt in 32-bit zu rechnen? Nun, Forth ist eine Sprache mit Schwerpunkt auf embedded control. Diese Controller haben im Gegensatz zu modernen PCs nur 4, 8, oder 16-bit breite Register. Alles was über diese Registerbreite hinausgeht kostet extensive Rechenzeit. F-PC und tCOM haben den Vorteil gegenüber 32-bit Systemen (so sehe ich das, Widerspruch muß natürlich hier kommen!), daß noch relativ hautnah an den typischen Forthproblemen (im Controllerbereich) geklebt wird. Umgekehrt gesagt: kompatible Forth-Quellen für PC und Controller zu schreiben, wird äußerst kritisch, wenn die Größe der CELLS unterschiedlich ist.

Noch zur Geschwindigkeit: Die folgenden Werte von einem 386DX 40 MHz zeigen, daß es durchaus Sinn macht, ökonomisch Befehle zu selektieren:

```
unsigned .32-/16-bit ... UM/MOD.....?2 µs
signed .....32-/16-bit ... M/MOD .....28 µs
signed .....32-/32-bit ... D/MOD .....169 µs
unsigned .64-/32-bit ... UMD/MOD .84 µs
```

```

33      ( drive -- total.clusters avail.clusters bytes/sec secs/cluster )
34      \ dl -- dx          bx          cx          ax
35      \ (current) default drive == 0 , A: == 1 , B: == 2 ....
36      \ if device NOT available ==> secs/cluster == 0
37      \ both F-PC and tCOM      F-PC only      tCOM only
38      \ wwwwwwwwwwww          wwwwww          wwwwww
39 &&F          pop dx
40 &&T          mov dx, bx
41      mov ah, # $36
42      int $21
43      cmp ax, # $FFFF          \ -1 ?
44      0= IF inc ax          \ -1 ==> 0
45      THEN
46 &&T          XCHG SI, SP
47      push dx          \ akg93nov20
48      push bx
49      push cx
50 &&T          XCHG SI, SP
51 &&T          MOV BX, AX
52 &&T          RET
53 &&F          1push
54      end-code
55
56 "" VALUE thousands-sep \ a separator for thousands millions ...
57
58 : (UD.) ( UD -- a | ) \ converts UNsigned DOUBLE using thous. sep.
59      $A save!> base
60      <# 12 1
61      DO # 2dup D0=
62      ?LEAVE
63      | 3 mod 0=
64      IF thousands-sep hold
65      THEN
66      LOOP #>
67      restore> base ;
68
69 : .MB ( d -- ) \ display MegaBytes in two ways          \ akg93nov20
70      2dup $0400 ( 1024 ) mu/mod
71      $0400 ( 1024 ) um/mod
72      4 .r ." MB " 4 .r ." KB ( = " drop          \ akg93dec02
73      (UD.) 13 over - spaces type ." Bytes )" ;
74
75 : (DRIVE?) ( -- n ) \ where are we? A==1 , B==2          \ akg92oct24
76      0 25 BDOS 1+ ; \ now same numbers as ?drive.extract ?drive.prepend
77
78 : .DRIVE ( -- )
79      (drive?) '@' + emit ." : " ;          \ akg92dec14
80
81 : diskspace ( drive -- D-total D-free per_cluster )
82      getdiskspace
83      * dup>R
84      um*
85      rot R@ um*
86      2swap R> ;
87
88 : .diskspace ( drive -- ) \ display info about 0 == default, 1 == A: ...
89 &&F 1 ?enough          \ some safety
90      cr ." Drive: "
91      dup 0=
92      IF ." default: " .drive
93      ELSE dup '@' + emit ':' emit
94      THEN
95      diskspace >R
96      \ dup 16 <>IF beep THEN          \ beep for <>1 MB free
97      2dup 1457664. [ 200 u2/ 10 + ] LITERAL R@ UM* D+ DU<>R          \ akg94jan12
98      \ compare to a 90mm HD-floppy with 200 files and 10 dirs
99      \ filled right to the brimm ( this is like the 'INSTANT-FORTH' disk)
100     cr ." free: " 2dup .MB
```

·UM/MOD und ·UMD/MOD sind reine CODE-Worte und daher relativ schnell gegenüber den COLON-Worten ·M/MOD und ·D/MOD .

Da in Forth und insbesondere in F-PC eine Unmenge optimierter Math-Operatoren bereitgestellt werden (dieser Sachverhalt hier mal positiv ausgedrückt!), bietet F-PC in der ak-Version ein Liste zur Auswahl im HyperText an. ·Help_Selecting... führt durch direkte Gegenüberstellung verwandter Operatoren zur zielstrebig richtigen Selektion. Früher ärgerte ich mich immer, daß diese Liste erst nach minimal drei Vormenüs erreicht wird. Ich benötige sie recht oft, so daß ich nun für die Mathematik Operatoren eine eigene FunktionsTaste kreiert habe: Shift-Ctrl-F1!

Wer mit FDISK (MicroSoft, auch IBM OS/2) öfter arbeitet, wird sich über die dort verwendete Division / Multiplikation / Rundung wundern. Auch dort wird nur mit 16-bit gekocht, aber versalzen. Es muß einer gewissen Anstrengung bedürfen, so grobschlächtige und inkonsequente Routinen zur Berechnung der Partitionsgröße zu programmieren. Digital Research (nun Novell) machte es ab spätestens v.5.0 besser.

'pictured' OUT and IN

Lange Zahlen mag ich schnell genug und insbesondere auch richtig auffassen können, daher die Trennung der Tausender durch ein Hochkomma. Diese von (guten) Taschenrechnern her gewohnte Schreibweise macht auch

```
101      R>
102      IF      ." !!! " beep      \ mark and beep, if less space
103      THEN
104      cr ." total:      " 2over .MB
105      cr ." used:      "  D- .MB
106      cr ." per Cluster: " R> . ." Bytes"
107      cr ;
108
109 &&F  0 .diskspace
110
111 &&F \S      ===== finito F-PC =====
112
113 : help-msg      ( a -- f )      \ print message for 'f'
114                                \ FALSE if no further action required
115      c@ 'f' <> dup
116      ?EXIT
117      cr ." usage:  dspace [drive1 [drive2] ..]"
118      cr ." like  dspace c e "
119      cr
120      cr ." DSPACE is a » tCOM « Utility by  akg93nov19--akg94jan12 " ;
121
122 : MAIN      ( -- )
123      bl word count      \ get char and count
124      IF      dup help-msg      \ display help ?!
125      IF      BEGIN      c@ UPC '@' - OMAX .diskspace
126                                bl word count      \ get next char and count
127                                0=
128      UNTIL
129      THEN
130      ELSE      0 .diskspace      \ no DOS-line option == default drive
131      THEN drop ;
132
```

Längste Zeile hat 99 Zeichen



mein DOS country.sys nach Patchen mit, was z.B. derNorton Commander vorteilhaft realisiert, und ich mich an internationalem (Datum und) Tausender freue. (Siehe auch Bemerkungen in 'NUN' in VD 4/92). In der Darstellung des FespPlattenfreiraumes müssen vorlaufende Nullen natürlich unterdrückt sein (bei mir sähe das sonst wie 0'000'031'000 Bytes aus). Dafür habe ich eine der besonderen Zahlendarstellungen aus F-PC-ak ausgewählt: ·(UD.'). Eine Unterdrückung der voreilenden Nullen und das Hochkomma nach jeweils 3-Stellen macht nur für ·Decimal einen Sinn, deshalb wird mit den sehr schnellen Operatoren ·save!> und ·restore> die ·BASE hin- und hergeschaltet. Der ·VALUE ·thousands-sep mag bei Bedarf auch auf Komma (USA) oder Punkt (Ger) oder spaces (sonstwo?) umgestellt werden, das alles führt aber international oft zu Problemen, das Hochkomma dagegen ist recht universell weil es typisch richtig interpretiert wird. Rein wie Raus: F-PC-ak akzeptiert das Hochkomma auch bei der Interpretation von Zahlen und erzeugt DOUBLES mit einer 0 in ·DPL (sollte ein DecimalPunkt folgen, wird DPL natürlich entsprechend gesetzt).

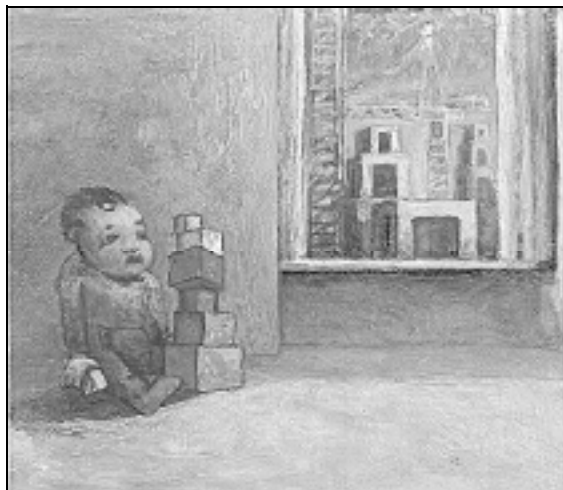
Auch die Clustergröße wird angezeigt, da sie indirekt immer für Verwirrung bis Bestürzung sorgt, insbesondere bei (blindem) Loslassen von msDOS auf moderne (große) Platten. Je Datei wird statistisch mehr als 50 % der Clustergröße an Kapazität verschenkt. (Bei 10'000 Dateien werden also gut 10 MB verschenkt, msDOS mag gerne gut 40 MB verschenken). Daher sind Partitionsgrößen von knapp (!) 128 MB so beliebt, und deshalb speichern 1M44 Scheiben mehr (!) als das Doppelte einer 720 kB Scheibe, und deshalb nützen die meisten DiskKomprimierer fast nichts bei vielen (kleinen) Dateien (wie z.B. bei F-PC, tCOM und vielen anderen Forth Systemen. Also vergessen Sie DoubleSpace und K.O. . Dagegen arbeiten pkZIP und lhARC u.a. intern ungerastert (Cluster-, Sector-frei). Neben den Kompressionsraten von 1:3 für typische Forth *.seq und 1:5 für Blöcke spart man dort je Datei zusätzlich nochmals diese halbe Clustergröße.

Die gewählte Art der Ausgabe soll der MegaBytigen Dreifaltigkeit entgegenwirken. Es ist KEIN Fehler, wenn dort einmal 31 MB 1023 KB ausgegeben wird! Technischwissenschaftlich bedeutet Mega == * 10^6. Unter Programm-Schaffenden wird darunter aber auch gerne * 10^3 KB verstanden (* 1'024'000) oder -- wenn schon, denn schon -- auch * 1024 KB = * 1'048'576. Da nach der erten Definition eine Festplatte scheinbar bzw. konkret in DM-Scheinen 4.633 % billiger ist als in der letzten Version,

Zusätzlich werden ein Beep und drei AusrufeZeichen ausgegeben, wenn der Platz für den Inhalt einer mit vielen Dateien gefüllten 1M44-Floppy nicht mehr ausreicht. Es reicht hier aber nicht ein simpler Vergleich mit 1.44 MB. Die Clustergröße wird noch mit berücksichtigt. Als Orientierung diene die Instant Forth Disk des F-PC-ak, die fast 200 Files in 10 Directories enthält.

Ich wünsche viel DiskSpace (allerdings nicht allein durch Programmierfehler, soll ja vorkommen).

□



Das müssen wir analysieren, da müssen wir uns äußern ...

I.*: Ja, also, was sagt Ihr denn zu dieser Diskussion um die Profis und die Hobbyisten und das? Da müssen wir uns doch äußern.

E.*: Quatsch, da halten wir die Klappe, die machen uns doch zur Schnecke, das hab' ich im Urin!

Ü.I.*: Warum fängst Du nicht mal an, im Kopf zu denken! Aber recht hast Du schon.

I.: Ey, Leute, denkt dran, daß der Rekorder läuft! (... *Füßescharren, betretenes Schweigen, Räuspern* ..) ... also ich finde, wir sollten das Problem angehen.

E.: Grade Du! Du laviest doch nur. Als dieser Firmenfritze neulich 'ne glatte halbe Stunde lamentiert hat - und das auf meine Telefonkosten - daß seine 20-DM-Anzeige nix bringt, dem hätte ich ja am liebsten...

Ü.I.: Erstens zahlen wir die Kosten zusammen und zweitens ist dieser Feierabend-Forthler, der immer meckert, daß seine Ascii-234 nicht korrekt ankommen, auch nicht besser.

I.: Aber der macht das immerhin per Mail. Ich finde, wir müssen die beiden Seiten irgendwie zusammenbringen.

E.: (*betriibt*) Dann sind wir überflüssig!

Ü.I.: (*erfreut*) Ja, dann sind wir überflüssig!

I.: Aber im Ernst: Der Versuch, diese angebliche Mehrheit von akademischen Forthlern als Bindeglied zwischen die Fronten zu schieben ...

E.: .. ist wohl voll in die Hose gegangen.

I.: Weil Du Deine blöden Depressionen grade ausgetragen hast und wir alles alleine machen mußten!

Ü.I.: Nee, da war er mal unschuldig. Die Idee war zwar logisch, aber die *academia* hat, wie üblich, praktisch versagt. Hat sich in ihren Elfenbeinturm des 'Sowohl, als auch, aber ich doch nicht!' zurückgezogen.

I.: Wenn's ernst wird, sind sie alle weg. Der einzige Lichtblick bisher waren die Kurzmeldungen.

Ü.I.: Seitdem ist der Spruch 'Leider bin ich grade ungemein beschäftigt!' praktisch ausgestorben.

E.: Was das an Telefonkosten spart.

I.: Und der Spruch kam ja nun von allen drei Seiten. Von den Profis kam ja fast nichts anderes.

Ü.I.: Aber seitdem reden die gar nicht mehr mit uns.

I.: Und zumindest bei der CD-Sache hätte ich schon gern wen gehabt, der das Organisatorische mal eben seiner Sekretärin aufdrückt.

E.: Mir wär's lieber gewesen, so eine Chaotentruppe zu haben, die sich nächtelang in Kneipen die Köpfe heißredet. Gefühl und Härte! Wißt Ihr noch damals, wie nach der Ini-Sitzung diese total hübsche ...

Ü.I.: (*lacht*) Ja, das war nett, aber in diesem Forth-Verein gibt's keine Frauen.

I.: Und Kneipentermine in Fernzone 3 gehen sowieso nicht. Aber es war ja auch keine Zeit da. Zumindest nicht für professionelle Arbeit. Wir torkeln hier immer nur von einer Notlösung zur

nächsten.

Ü.I.: Und für basisdemokratisches Entscheidungs-Gelaber reicht die Zeit schon gar nicht.

E.: Dann hätten wir's halt verschoben und richtig professionell gemacht...

I.: Das Kneipenflirten?

Ü.I.: Blödmann! (*grinsend*) Du weißt doch, wie schwer ihm das fällt.

E.: Na, ja, letztenendes macht's mir auch total Spaß, Sachen mal eben so hinzupfuschen, für die andere Leute Jahre brauchen.

I.: Dazu sind wir schließlich Forthler! Und was dabei rauskommt, ist verflucht gut. Aber wie machen wir jetzt weiter?

Ü.I.: Also, dieses Bild, diese Collage sollten wir schon mit reinsetzen: Der Editor beim Tanz auf den Eiern, zwischen Schwerindustrie und Schachspielern und unter ihm die wegbrechende Universitas.

E.: Aber die Ruine von der TU 1945 erkennt doch kein Mensch!

I.: Leute, die Spalte ist gleich zu Ende und ich will ins Bett! Wie machen wir weiter...

Ü.I.: Na, so wie bisher, wie denn sonst!

E.: Klar, und es macht ja auch immer wieder Spaß.

I.: Ok! Dann drucken wir das mal so..

Dieses round-table-Gespräch des Editorenkollektivs aus aktuellem Anlaß wurde aufgezeichnet und redigiert von Claus Vogt.



Rätsel

benütztes System:

Tom Zimmers ZF auf
PC XT --> Pentium



Der Edgar Prellwitz meinte einmal bei einer Diskussion in der FORTH-Gruppe Moers, (es ging um WIN32FOR und Windows95 .bzw Win32s) man könne doch einfach mal mit einem Disassembler Win32s analysieren!, wenn man wissen wolle, was es denn so mache.

Ich bin meinerseits davon überzeugt, daß es sehr schwer ist, Windows auf diese Art, ohne weiterführende Literatur zu verstehen und später zu programmieren. Ein disassembliertes Rohlisting, ohne symbolische Konstanten, Label usw. reicht allein kaum aus.

Um dem Edgar zu zeigen, wie schwer es sein kann, etwas zu analysieren, von dem man (1) nicht weiß, was es macht (2) nicht weiß, wie es das macht, habe ich das untenstehende kleine Programm geschrieben, das, da in Hochsprache formuliert und teilweise kommentiert, recht lesbar ist.

Es wäre einfach in Assembler zu implementieren, aber dann noch schwerer zu lesen und und zu verstehen.

Vielleicht hat ja der eine oder die andere aus der FORTH-Gemeinde Lust, sich ebenfalls an dieser Aufgabe zu versuchen.

Also: Was macht Krypt2 bei wiederholtem Aufruf (Vor dem ersten Aufruf von Krypt2 INIT aufrufen!)?

Und: Wie funktioniert das?

Ich persönlich finde den benutzten Algorithmus überaus nützlich, elegant und mit einfachsten Mitteln umsetzbar. Er ist von gleicher schlichter Eleganz, wie das Newtonsche Wurzelziehen.

Einen noch genaueren Algorithmus und die Lösungen mit Kommentar (soweit mein Verständnis dazu reicht!) finden Sie in der nächsten VD



Viel Spaß
M. Bitter 8)>

Martin Bitter

Möllenkampweg 1a; 46499 Hamminkeln (Mehrhoog)
Tel.: 02857/1419



```

1 32 Constant Max_stellen      \ so genau ist dieser einf.
2                               \ Algorithmus
3 32 Constant Stellen          \ soviel Stellen werden berechnet
4 0 Constant Uebertrag        \ Platz für den Übertrag
5
6 Create Field
7 Max_stellen 1+ 3 * 2* allot \ Das notwendige Array .
8
9 : I@ ( I -- n )              \ holt den Wert aus Position I
10 2* Feld + @ ;
11
12 : !! ( n I -- )            \ speichert n an Position I
13 2* Feld + ! ;
14
15 : fill_array ( -- )        \ füllt das Array mit Zweien
16 Max_stellen 3 * 0          \ Schleifenwerte
17 DO 2 I !! LOOP ;          \ jede Zelle mit Zwei vorbesetzen
18
19
20 : init ( -- )              \ richtet die Werte für ??? ein
21 fill_array 0 =: Uebertrag ;
22
23 : Summe ( I -- n )         \ Summe aus Uebertrag und Wert M
24 I@ 10 *                    \ m holen; mal 10
25 Uebertrag + ;              \ Summe bilden
26
27 : >Uebertrag ( I q -- )    \ bildet den neuen Wert für den
28 * =: Uebertrag ;          \ Übertrag
29
30
31 : krypt1 ( I -- q )        \
32 dup dup                    \ der Index wird dreimal gebraucht
33 Summe swap                 \ Summe bilden
34 2* 1+                      \ zugehörige ungerade Zahl bilden

```



```

35 /mod swap                  \ Rest und krypt1 ermitteln
36 rot !! ;                   \ Rest speichern; krypt1 TOS
37
38 : Schritt ( n -- )        \ Berechnet die Werte für eine Spalte
39 dup krypt1                 \ krypt1 berechnen
40
41 >uebertrag ;               \ neuen Übertrag merken;
42
43 : letzter_schritt ( -- )
44 0 Summe 10 /mod           \
45 . 0 !                      \ Ergebnis ausgeben
46 0 =: Uebertrag ;          \ Übertrag neu initialisieren
47
48 : krypt2 ( -- )           \ Hier ist das Rätsel:
49 Stellen 3 * 0              \ Was?
50 ?DO Stellen 3 * i - schritt \ Wie?
51 LOOP                       \ Wieso klappt's?
52 letzter_schritt ;         \

```



Forth-Gruppen regional

Berlin	Claus Vogt Tel.: +30 -782 81 79 p Treffen: nach Absprache
Rhein-Ruhr	Jörg Plewe Tel.: +208 -49 70 68 p Treffen: jeden 1. Samstag im Monat im S-Bahnhof Derendorf Münstererstr. 199,
Moers	Friederich Prinz Tel.: +2841 -5 83 98 p Treffen: jeden Samstag 14:00 Arbeitslosenzentrum, Donaustr. 1, Moers
Darmstadt	Andreas Soeder Tel.: +6257 -27 44
Mannheim	Thomas Prinz Tel.: +6271 -28 30 p Ewald Rieger Tel.: +6239 -86 32 p Treffen: jeden 1. Mittwoch im Monat, Vereinslokal Segelverein Mannheim e. V., Flugplatz Mannheim-Neuostheim

µP-Controller Verleih

Thomas Prinz
Tel.: +6271 -28 30 p

Gruppen Gründungen, Kontakte

Regional	Stuttgart Wolf-Helge Neumann Tel.: +711 -8 87 26 38 p
Fachbezogen	8051 ... (Forth statt Basic, e-FORTH) Thomas Prinz Tel.: +6271 -28 30 p

Forth-Hilfe für Ratsuchende

Forth allgemein	Jörg Plewe Tel.: +208 -49 70 68 p plewe@mpi-dortmund.mpg.de Karl Schroer Tel.: +2845 -2 89 51 p Jörg Staben Tel.: +2103 -24 06 09 p
------------------------	---

Spezielle Fachgebiete

Arbeitsgruppe Marc4	Rafael Deliano Tel./Fax.: +89 -841 83 17 pf
Anfänger und Wiedereinsteiger	Gerd Limbach Tel.: +2051 -25 51 12 p Mo.+Di. 20:00 - 22:00
32FORTH (Atari)	Rainer Aumiller Tel.: +89 -6 70 83 55 gp
FORTHchips (FRP1600, RTX, Novix...)	Klaus Schleisiek-Kern Tel.: +40 -330 674 p
F-PC & TCOM, ASYST (Meßtechnik), embedded controller(H8/5xx//TDS2020, 8051 ... eFORTH...), FUZZY	Arndt Klingelberg Tel.: +2404 -6 16 48 agp
Gleitkomma-Arithmetik	Andreas Döring Tel.: +721 -59 39 35 p
HS/Forth (Harvard Softworks)	Wigand Gawenda Tel.: +30 -44 69 41 p
KI (Künstliche Intelligenz), OOF (Object Oriented Forth)	Ulrich Hoffmann Tel.: +4351 - 712 217 pf Birgit Steffenhagen Tel.: +38204 - 129 33 pa +381 - 498 35 52 g
Forth-Vertrieb	volksFORTH/ultraFORTH, RTX/FG/Super8/KK-FORTH Ingenieurbüro Klaus Kohl Tel.: +8233 -3 05 24 p Fax: +8233 -99 71 f
Forth-Mailbox (KBBS)	+431-533 98 98 :8N1 Sysop: Holger Petersen Fax: +431-533 98 97 f Tel: +431-533 98 98 p <i>bis 22 Uhr</i> Mail: hp@kbbs.org

Hinweise

Zu den Telefonnummern

f == FAX
a == Anrufbeantworter, hier können Sie Ihren Ansprechpartner eventuell vorinformieren,
erwarten Sie bitte keinen (kostspieligen) Rückruf
g == geschäftlich, zu erreichen innerhalb typischer Arbeitszeiten
p == privat, zu erreichen außerhalb typischer Arbeitszeiten

Die Adressen des Forth e. V. (Forth Büro) und der Redaktion/ finden Sie im Impressum

FORTECH Software

- Forth-Entwicklungsumgebung comFORTH unter DOS oder Windows
- interaktive Crossentwicklungssysteme für Mikroprozessoren von Intel, Motorola, Zilog, TI ...
- Softwareentwicklung für PC und Mikrocontroller
- System- und Anwendungsprogrammierung unter Windows

comFORTH

- Forth-Entwicklungsumgebung für Windows
- interaktive Benutzbarkeit aller Windows-API-Funktionen und -Strukturen
- kombinierbar mit anderen Programmiersprachen
- Unterstützung von DDE, DLL, VBX, ...

fieldFORTH

- Forth-Entwicklungssystem für eingebettete Systeme
- interaktive Programmierung off-line und on-line
- verfügbar für diverse 8-, 16- und 32-Bit Mikrocontroller und -Prozessoren (TMS320C40, M68332, M68HC11,...)
- **NEU!!!** Evaluation-Kit M68HC11 inclusive Board MINI-HC11 296,70 incl. MwSt.

FORTECH Software GmbH

J.-Jungius-Str. 9 • D-18059 Rostock • Tel: (03 81) 4 05 94 72 • Fax: (03 81) 4 05 94 71