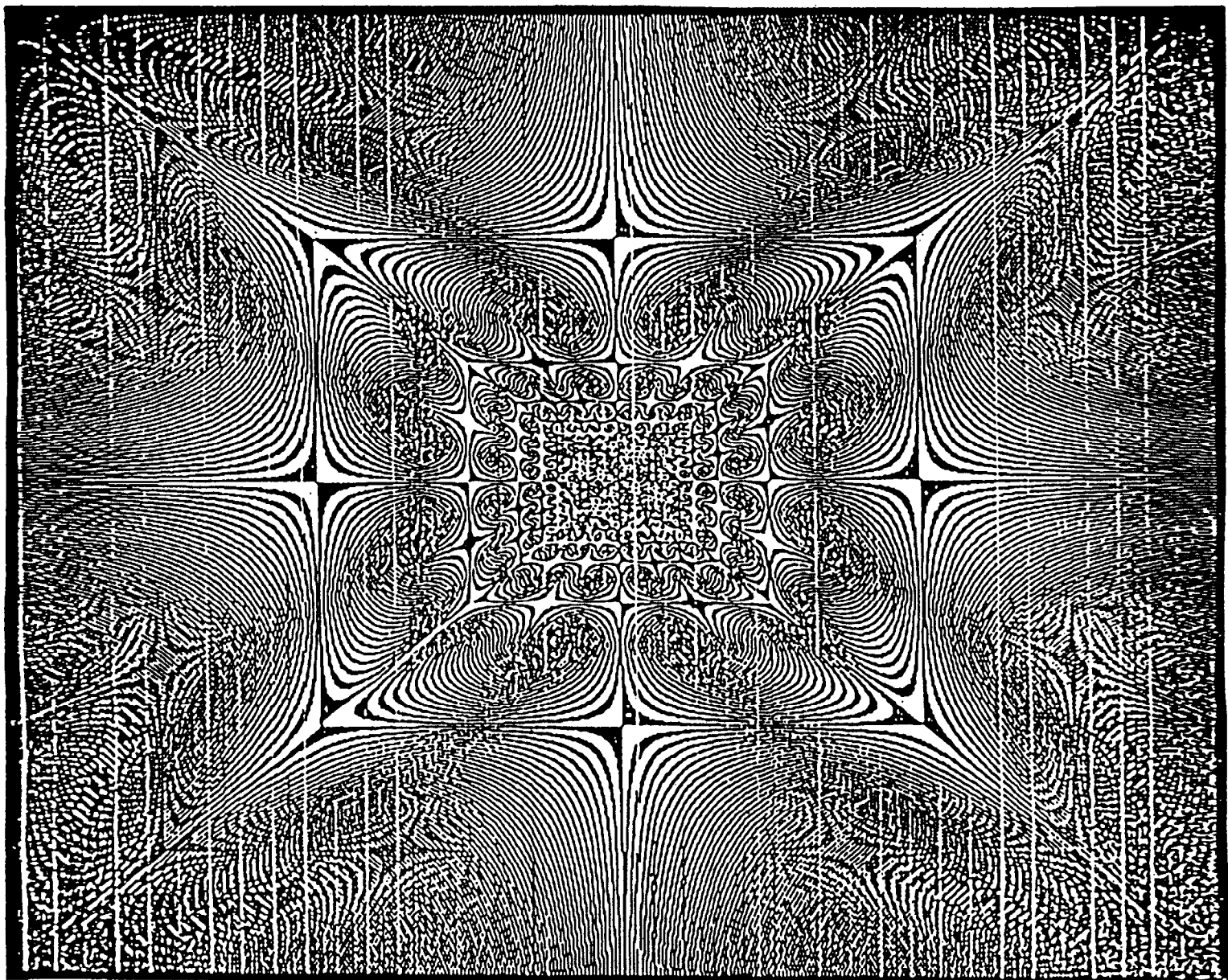


Vierte Dimension

mai 84



***** INHALTSVERZEICHNIS *****

EDITORIAL / IMPRESSUM	SEITE 2
INTERVIEW MIT BILL RAGSDALE	SEITE 3
ARBEITSGRUPPE LEIBNITZ	SEITE 4
ARBEITSGRUPPEN 4.DIMENSION / DOKUMENTATION	SEITE 4
LESERSEITE	SEITE 5
BUCHBESPRECHUNG (R. ZECH 1984)	SEITE 5
ARBEITSGRUPPE 5.DIMENSION	SEITE 6
SOFTWARE - GILDE	SEITE 6
EIN GANZ EINFACHER FALL	SEITE 8
BEZUGSQUELLEN FÜR FORTH - INSTALLATIONEN	SEITE 12
FRAGEDOGEN 1/84	SEITE 15
EINSTIEG	SEITE 16
UNTERSTÜTZUNG	SEITE 16

EDITORIAL

Seit gut 8 Wochen gibt es die **Forth-Gesellschaft**, als erste offizielle westdeutsche Vertretg. (Chapter) der FIG (Forth Interest Group) in den USA. Unser Kommunikationsorgan, **VIERTE DIMENSION** genannt liegt Ihnen vor. Dieser Newsletter sowie die Forth - Gesellschaft haben es sich zur Aufgabe gemacht allen Forth-Programmierern (und denen die es werden wollen) in Deutschland Unterstützung zu bieten, ihnen Beratung und Informationen zukommen zu lassen, sowie ihnen darüber hinaus die Möglichkeit zu geben Erfahrungen untereinander auszutauschen. Zum einen wird dies über diesen Newsletter geschehen, zum anderen aber auch über neue Ortsgruppen in der BRD. Jeder der also Kontakt zu Gleichgesinnten sucht, ist hiermit eingeladen, eine eigene Ortsgruppe zu Gründen. Informationen über die Art und Weise, sowie Unterstützung beim Aufbau geben wir Ihnen gern !

In dieser ersten Ausgabe (die obligatorische Nullnummer) versuchen wir zunächst auf die Mengen an Anfragen einzugehen, die wir hier empfangen haben. Die Leserseite, sowie das sehr lange Interview mit Bill Ragsdale, dem Begründer der FIG sind beredetes Beispiel dafür. Im groben und ganzen ist die Beschreibung von Bill auch für die Forth - Gesellschaft gültig. Aber wir planen auch darüber hinaus aktiv zu sein. Der Artikel von Andreas Goppold läßt schon einiges ahnen. Hierin wird auch das Grundprinzip unserer Gesellschaft erläutert: **FEEDBACK** oder zu Deutsch:

VON NICHTS KOMMT NICHTS !

Will meinen: Die Forth - Gesellschaft ist nicht etwa ein Haufen abgedrehter Techniker o.ä., nein die Forth - Gesellschaft das sind SIE, ist jeder Einzelne von uns, der seinen Beitrag zur Entwicklung leistet. Und sei es indem er diesen Newsletter abonniert. Die Fragebögen am Ende des Heftes geben auch Auskunft über andere Ansatzpunkte. Dieser Newsletter zum Beispiel wäre ohne die freiwillige Mithilfe einiger Personen gar nicht zustande gekommen. Da wären zu Beispiel K.Schleisiek und A.Goppold, die sich beide unheimlich ins Zeug gelegt haben, damit dieses Heft auch interessant wird. Oder die **Spender** (A.Goppold und R.Löhr sowie ein unbekannter Spender) die uns mit Sach & Geldspenden erst die Perspektive vermittelt haben, daß Forth (und damit die Forth - Gesellschaft) eine Zukunft hat. H.Jochen Dahmke hat uns sehr geholfen, indem er uns einen vernünftigen Drucker zur Verfügung stellte. Übrigens: Jochen ist als Leiter der **AG - Dokumentation** auch sehr auf unser aller Mithilfe angewiesen. Wer kennt das nicht: Eine bestimmte Information, von der man schon einmal gehört hat, die einem sehr viel Arbeit ersparen könnte ist einfach nicht aufzutreiben. Die **AG-DOKU** will und wird hier Abhilfe schaffen.

Ein weiteres Thema sind Utilities. Mit dem CASE? - Artikel wollen wir eine Serie einleiten, die dem Anfänger (vielleicht auch manchem Semi-Profi) ein bisschen die Tricks aus der Forth-Kiste zeigt und erläutert. Also: Wer selber gute Ideen hat, kann diese über die **VIERTE DIMENSION** vielen Leuten zugänglich machen. Denn das ist das Bestreben der Forth-Gesellschaft: Gute public domain - Software unter die Leute zu bringen. (S. auch **AG-Leibniz!**)

Die **VIERTE DIMENSION** ist also nicht etwa die 1024ste Computerzeitschrift, sondern sie ist das Organ eines Netzwerkes welches sich aus den aktiv an Forth Interessierten Personen, also **EUCH / IHNEN** konstituiert.

Apropos: Netzwerk, die kurz vor Redaktionsschluß ins Leben gerufene **AG-Modem** wird zu diesem Thema wohl auch noch einiges verlauten lassen.

Ein letzter Hinweis in eigener Sache: Als **NON-PROFIT** Organisation sind wir natürlich auf mög-

lichst viele Mitglieder und Sponsoren angewiesen. Unser recht geringer Mitgliedsbeitrag geht zum größten Teil in die Produktion dieses Newsletter. Wenn Sie also spezielle Anfragen haben, die wir natürlich gern beantworten, legne Sie bitte 3,- DM in Briefmarken bei & einen Rückumschlag. Damit wir genügend Spielgeld für erforderliche Recherchen und die Antwort an Sie haben. Wie gesagt: Von Nichts kommt Nichts.

Und nun viel Spaß

MAY FORTH BE WITH YOU !

IMPRESSUM :

VIERTE DIMENSION, Kommunikationsorgan der Forth - Gesellschaft (FIG - Deutschland)

Herausgeber & Verantwortlich:

Horst - Günter Lynsche
Holstenstrasse 191
2000 Hamburg 50

Redaktion: K.Schleisiek, Thomas Braun (Layout)
A.Goppold, H.Jochen Dahmke & Eckart
W.F. Schmidt (Druck).

Namentlich gekennzeichnete Artikel geben nicht unbedingt die Meinung der Redaktion wieder. Nachdruck, auch auszugsweise nur unter Quellenangabe gestattet. Unverlangt eingesandte Manuskripte werden ausnahmslos veröffentlicht, so der Platz reicht.

Erscheinungsweise: ca. 4 * pro Jahr
Telefonische Anfragen: Werktags von 12 - 19 Uhr
sowie Einsendungen an :

COMMON INTERFACE ALPHA

HOLSTENSTRASSE 191

2000 HANBURG 50

040 43 50 70

FORTH?

Forth wurde von Charles Moore über einen längeren Zeitraum hinweg entwickelt, als er sich als Programmierer mit den unterschiedlichsten Maschinen und Programmiersprachen herumschlagen mußte. Während seiner Arbeit Anfang der 70er Jahre in Kitt Peak, einem Radioastronomieinstitut in den USA, hatte sich ein System entwickelt, daß alle Strukturmerkmale von Forth, so wie wir es heute kennen, enthielt. Er wollte dieses Entwicklungssystem **Fourth** nennen, in Anspielung darauf, daß damals alle Welt von der dritten Generation von Computern sprach. Der zur Übersetzung verwendete Fortran Compiler gestattete aber nur Namen von maximal fünf Buchstaben - das u mußte gehen und es blieb Forth übrig. Es dauerte nur wenige Jahre, bis Forth die Standardprogrammiersprache der Radioastronomen geworden war. Von hieraus verbreitete sich dann Forth als eine Art mündlicher Tradition von Programmierer zu Programmierer weiter, bis dann im Jahre 1978 die Forth Interest Group gegründet wurde. Siehe dazu auch das Interview mit ihrem Begründer, B. Ragsdale, in diesem Heft.

EIN INTERVIEW MIT BILL RAGSDALE

Ich habe mir erzählen lassen, wie die Forth Interest Group anfang, aber ich konnte nie herausfinden, warum ...

"Warum"? Es gibt kein Warum. Es ist einfach so passiert und fing einfach an zu wachsen. Es gab keinen Grund als die Erkenntnis, daß ein allgemeines Bedürfnis existierte, Informationen auszutauschen. Das "Wie" geschah folgendermaßen: 1977 oder '78 hielt ich vor einem Computerclub einen Vortrag über höhere Sprachen und strukturierte Programmierung. Dabei benutzte ich Forth als Beispiel. Damals hatte ich gerade ein sehr primitives Forth an meinem Arbeitsplatz zum Laufen gebracht und war ziemlich stolz darauf. Ich startete es und zeigte es den Leuten. Ich mußte einen Lochstreifen laden, Forth meldete sich mit "ok" und lief äußerst eingeschränkt.

John James und Dave Boulton waren bei diesem Treffen anwesend. John hatte vorher noch nie von Forth gehört und ihn interessierte es philosophisch - es hatte genau die Eigenschaften, die ihm wichtig erschienen. Dave nutzte es schon professionell als Angestellter einer Firma in San Francisco. Also kamen wir drei ins Gespräch und einigten uns, uns später zu einem vertiefenden Gespräch zu treffen. Dann schlug noch irgendwer vor, daß ich einen Vortrag vor dem Homebrewcomputerclub geben sollte. Nach diesem Vortrag kamen diejenigen, die sich dafür interessierten, zusammen, und wir trafen uns unter einer Eiche in der StanfordMensa. Kim Harris, Dave Kilbridge, John James und Dave Boulton waren anwesend. Wir waren damals die Gründer von FIG. Einige andere waren auch da; einer - Tom Olsen - hatte Forth auf einer PDP-11 zu Hause im Wohnzimmer, vollgepackt mit drei Floppydrives, Harddisk, Bildschirm und Drucker.

Wir beschlossen, uns wieder zu treffen und unsere gemeinsamen Interessen herauszuarbeiten. Dies geschah ungefähr einen Monat später. Ich erinnere mich, daß es der Super Bowl Sonntag 1978 (Football Endspiel) war, als FIG in einer Wohnung in Sunnyvale gegründet wurde. Damals haben wir geschworen, einen Rundbrief herauszugeben. John James hat die ersten Artikel geschrieben, und fünf oder sechs aus unserer Gruppe haben einen Leitartikel entworfen; damit entstand Forth Dimensions. Tom hat die ersten Ausgaben auf

seiner PDP-11 bearbeitet. Zu dem Zeitpunkt übernahmen meine Frau Anne und ich den Druck, die Herausgabe und Versendung von Forth Dimensions. Wir sammelten das Material und organisierten die nächsten Ausgaben. Unglücklicherweise starb Tom zu der Zeit an Diabetes. Damit verloren wir gleich zu Beginn eines der Gruppenmitglieder, und wir restlichen fünf machten weiter.

Drei Monate später war die West Coast Computer Faire. Damals hatten wir einige Ausgaben von Forth Dimensions und wir verschenkten sie - die ersten drei Ausgaben waren kostenlos, weil wir keine Vorstellung über die Lebensfähigkeit unserer Aktivitäten hatten. Durch die Computer Faire bekamen wir ca. 200 Leute als Mitglieder und Bezieher. Im darauffolgenden Jahr ging es dann richtig los.

Wie habt ihr am Anfang Informationen über Forth verbreitet?

Eines Samstags haben wir angekündigt: "Es findet eine Schulung über Forth bei Dorado Systems um zehn Uhr statt". 30 Leute kamen zusammen. Wir hatten den Raum voller Leute. Jeder Stuhl, jeder Sessel und der ganze Boden war voller Menschen. Dann hielten wir einen Vortrag über Forth. Ich hatte ein kleines Demonstrationssystem zur Verfügung, so daß sie ein minimales System wirklich laufen sehen konnten. Wir haben die Schulung einige Stunden gemacht und fragten dann, wer ein laufendes Forth System hat. Von den 30 hoben nur drei ihre Hand, soweit ich mich erinnere: Ich selber, Dave Boulton und Tom. Ich war erschrocken, weil wir wußten, daß vom Lerneffekt her die Leute sich nach einer Woche kaum noch an das erinnern würden, was sie gelernt hatten.

Bald danach - es könnte bei diesem Treffen gewesen sein - wurde die erste Forth Implementation auf einem Mikrocomputer für einen PET verfügbar. Bei Forth DIMENSIONS kamen sofort Briefe an, die sich darüber beklagten, wie schlecht das Produkt war und wie sie es hassten. Das hat uns beunruhigt. Da war ein Anbieter, der laut hinausposaunte, daß er ein Forth System für einen PC hatte, das sehr schlecht war. Inzwischen hatten wir Kontakte zu Elizabeth Rather von Forth Inc. und wir erkundigten uns, was sie mit PC's vorhätten; würden sie Forth für Apples und Pets herausbringen?

Sie sagte sehr deutlich "Auf

keinen Fall"; dies sei nicht ihre Mission und sie würden keine Unterstützung für PC's geben. Sie verkauften Systeme, die 2000 - 10000 Dollar kosteten für wissenschaftliche und industrielle Anwendungen.

Die Historie hat dies bestätigt, obwohl sie nun endlich nach fünf Jahren eine Version für den IBM-PC herausgegeben haben. Zwischendurch aber haben sie nie ein kommerzielles Produkt für einen PC vermarktet. Diese beiden Faktoren - einerseits die Möglichkeit, daß Leute, die es nicht besser wissen, furchtbar schlechte Implementationen herausgeben, und andererseits die Aussage unseres Marktführers, daß sie PC's nicht unterstützen werden - bedeuteten, daß wir große Schwierigkeiten haben würden, Forth zu lehren und zu erlernen. Es würde keine kommerzielle Quellen für die Sprache geben.

Also beschloß FIG, seine eigene Implementation zu machen?

Nicht ganz. Wir wollten Richtlinien herausgeben an geeignete Verkäufer, aber es gab keine. Da beschloß ich, "mit dem Speck nach dem Schinken zu werfen". Zu dem Zeitpunkt hatte ich bereits die dritte Version unseres Systems von Dorado Systems herausgegeben. Es war bei Version 3.6 angekommen und so langsam zu einem brauchbaren System geworden. So beschloß ich nach sorgfältiger Überlegung, die Version 3.6 freizugeben und auf mein Copyright zu verzichten; dies wurde die Grundlage des FIG Modells. Tatsächlich mußte ich mir das neulich nochmal angucken, und wenn Sie sich ungefähr block #54 des FIG Modells ansehen, dann steht da folgende abort-Meldung: "Forth 65 version 4.0". Forth 65 war unser interner Name dafür, version 4.0 war ein bereinigtes 3.6 und wurde fig-FORTH 1.0.

Mit der Freigabe dieses Systems brauchten wir mehrere, portierbare Implementationen. Unseres war für den 6502, und ich war weder von technischen noch zeitmäßigen Gesichtspunkten her in der Lage, Versionen für sechs oder acht andere Prozessoren zu schreiben. Deshalb veröffentlichten wir in Forth Dimensions einen Aufruf, daß diejenigen, die Assemblerprogrammiererfahrung und Interesse an Hochsprachen haben, sich zu einem Workshop einfinden sollten.

Wir bekamen eine Menge Resonanz und filterten für jeden Prozessor zwei Leute aus:

(Forts. Seite 7)

Die Arbeitsgruppe Leibniz

Eine große Schwierigkeit, den Umgang mit Computern selbstverständlich zu machen, liegt darin, daß "Computerese" eben zunächst einmal Englisch ist. Konkret bedeutet dies, daß jedem, dem Englisch nicht geläufig ist - und das sind auf die Gesamtbevölkerung betrachtet eben die meisten - eine zusätzliche Hürde vor das Computerverständnis gebaut wird. Das ist zum Beispiel auch ein Punkt, an dem N. Wirth mit einer "jeder-Akademiker-kann-Englisch-Attitüde" mit der Definition von Pascal versagt hat.

Kurz: Die Arbeitsgruppe Leibniz hat es sich zur Aufgabe gemacht, den deutschen Sprachraum mit einer Programmiersprache zu beglücken, die außerdem noch auf eine Eigenart der deutschen Grammatik eingeht, nämlich ihre RPN Struktur.

(RPN = "Reverse Polish Notation"). Einige werden es inzwischen schon ahnen: Es soll ein getarntes Forth entwickelt werden. Aufbauend auf der Syntax und der Semantik des Forth-83 Standards, sollen schlicht die englischen Worte durch deutsche ersetzt werden.

Folgende Nachteile/Schwierigkeiten zeichnen sich schon am Horizont ab:

Nach aller Erfahrung ist mit mehr Tipparbeit zu rechnen, da in der Regel dt. Worte länger sind. Und im dt. Zeichensatz gibt es weder den "Klammeraffen" § (und welcher anarchische Forth Programmierer will schon mit dem "Paragraphenzeichen" zu tun haben !?) noch etwa eckige, geschweifte denn geschweifte Klammern.

Dies wäre also die erste Phase der Arbeitsgruppe: Die Bestimmung der dt. Namen. Dies ist nicht einfach, läßt sich auch nicht erzwingen und benötigt meist einen total nicht-deterministischen Geistesblitz - kurz, blödeln ist gefragt und dies ist die große Stunde der Spinner.

In der zweiten Phase werden die so gewonnenen Worte in existente Systeme gesteckt und Metacompiliert - dann müssen nur noch die üblichen Tools kurz umgeschrieben werden, was sich aber leicht automatisieren läßt. Dieses resultiert in einer Modellimplementation.

In der dritten Phase muß dann dieses Modell nicht nur auf die verschiedensten Prozessoren gehievt werden, nein auch noch auf die verschiedensten Maschinen und I/O Interfaces. Dies ist sicher nur dadurch möglich, daß sich kompetente Programmierer finden, die diese Arbeiten außerhalb

der Arbeitsgruppe im engeren Sinne übernehmen.

Was ich noch nicht erwähnt hatte: Leibniz wird ein public domain-produkt sein. Witzigerweise gibt's dafür keine Übersetzung außer der, daß der Autor auf sein Copyright verzichtet und halt ausdrücklich gesagt wird, daß so oft kopiert werden soll, wies nur geht. Dies erscheint mir als die einzig mögliche Strategie, gegen den versammelten Schwachsinn der Basics und Pascals anzukommen. Wenn es uns gelingt, Schülern durch Leibniz auf ihren VC-XX und ähnlichen Krüppelcomputern klarzumachen, was programmieren ist, dann sind sie auf ewig für Basic und verwandte Geistesfolterwerkzeuge verloren.

K. Schleisiek

AG - VIERTE DIMENSION

Die Arbeitsgruppe VIERTE DIMENSION beschäftigt sich im wesentlichen mit der Vorbereitung und Produktion des gleichnamigen Newsletters. Sie besteht bisher aus 6 Personen, zwei Haupt-Ehrenamtlichen Redakteuren und (bisher) und vier freien Mitarbeitern. Die beiden Redakteure sind vollauf mit der Erfassung und Gestaltung der Texte beschäftigt, zwei der Freien Mitarbeiter übernehmen die Endproduktion. Alle anderen Freien Mitarbeiter (wie gesagt: bisher zwei) produzieren Texte, Meldungen oder geben auch einfach mal eine Information zum besten. Von den letzteren können wir eigentlich gut noch mehr gebrauchen, und freier Mitarbeiter werden ist gar nicht schwer: Man nehme eine Idee, verhackstücke sie in einen leserlichen Text, sende diese an die Redaktion der VIERTE DIMENSION und mit ein bisschen Glück ist man in der nächsten Ausgabe zu lesen. Wobei, am liebsten sind uns ja mit Wordstar erstellte Texte auf Diskette in CP/M - Standard-Format, besser noch in unserem sprich: ITT DD/DS - Format. Aber natürlich nehmen wir auch Manuskripte entgegen. WAS bzw WER uns bis jetzt noch völlig fehlt ist die Hilfe eines bzw ein Grafikers / Zeichners. Abhilfe ist herzlich willkommen. Das jedoch nur Nebenbei bemerkt.

Im Common Interface Alpha, finden auch noch andere Dinge statt, zum Beispiel steht der Kopierer den wir zur Vorproduktion des Newsletters benutzen, natürlich auch für Lis-

tings, Artikel, Dokumentationen usw zur Verfügung. Wer also in der Liste der Arb.Gem. Dokumentation einen interessanten Artikel bzw ein interessantes Programm findet kann sich dieses für 0,15 DM pro Blatt durch uns kopieren lassen. Das Thema "Vernetzung" durch das C.I.Alpha wurde bereits Eingangs erwähnt. (Siehe auch Fragebogen!)

Und last not least: Im und durch das C.I.Alpha wird auch in FORTH ausgebildet !! Wir haben verschiedene Kontakte zu Personen die für einen akzeptablen Preis in FORTH ausbilden. Und das auf verschiedenen Ebenen: Einmal gibt es da den Kursus für " Systemprogrammierer" und ähnlich hohe Tiere, also Leute die schon recht gut Bescheid wissen (sollten) und eigentlich nur noch die letzten Tricks und Kniffe lernen wollen. Die Expertenebene, dann gibt es die Kurse die für Computer-Hobbyisten oder FORTH - Neulinge gemacht werden. Hier wird schon ein grundlegendes Verständnis der Funktionsweise eines Micro-Computers vorausgesetzt und die Sprache FORTH als Instrument zur Beherrschung desselben gelehrt. Und dann gibt es noch die Basis-Ausbildung, für alle die bisher gar nichts oder wenig mit Computern im Sinn hatten und die natürlich auch noch kein FORTH kennen, aber beides lernen wollen. Und erstaunlicherweise, ist die letzte Gruppe die Gruppe, die es am einfachsten hat FORTH zu lernen. Denn das Experiment in einer Kleingruppe von 7 Personen hat es bewiesen: Computerneulinge lernen INDEM sie FORTH lernen :

1. Den Computer viel natürlicher kennen und besser zu beherrschen und
2. die Sprache FORTH schneller und auch besser anzuwenden als Leute die (wie der Autor) durch Basic oder ähnliches (letztlich durch veraltete Konzepte über den Computer bzw die Sprachen) "verdorben" wurden. Vorerst sind diese Kurse als Angebot gedacht. Das heißt erst bei genügend Interesse werden sie regelmäßig stattfinden. Die genauen Termine / Örtlichkeiten / Gebühren stehen noch nicht fest, sind aber im C.I.Alpha zu erfahren. Bisher ist dieses Angebot auf den Norddeutschen Raum beschränkt, d.h. für den Raum Hannover haben wir auch schon eine Kontaktadresse parat. Über weitere Kontakte in anderen Regionen würden wir uns sehr freuen.

LESERSEITE

Um auf die, in dem Berg von Einsendungen, am häufigsten an uns gestellte Frage einzugehen, wird an dieser Stelle auf einen Beispielbrief geantwortet.

H.J.Burg aus Vlotho schrieb uns :
Ich suche eine wirklich gute Einführung in FORTH. Obwohl ich einigermaßen englisch spreche, wäre mir eine deutsche Einführung in FORTH, wegen der vielen Fachausdrücke lieber. Gibt es "Starting Forth" schon in deutsch ? Gibt es einen Decompiler, mit dem man die FORTH-Worte wieder in ihre einzelnen Ursprungsworte zelegen kann, um so etwas mehr über FORTH zu lernen?

Eine deutsche Version des Einführungskurses "Starting Forth" von Leo Brodie erscheint in Kürze bei einem Münchener Verlag unter dem

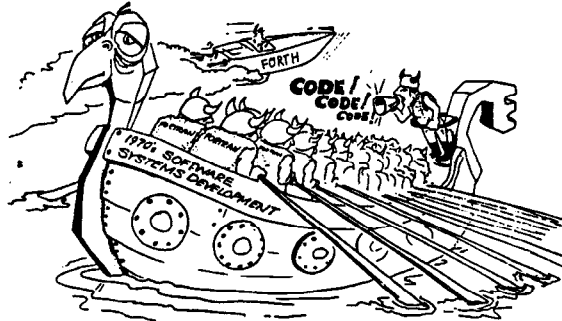
Namen "Programmieren in FORTH". Ein anderes, sehr gutes Buch zum Thema FORTH in deutscher Sprache, das sowohl für Anfänger mit Erfahrungen aus anderen Programmiersprachen als auch für den fortgeschrittenen FORTH-Programmierer geschrieben wurde, ist "Die Programmiersprache Forth" von Roald Zech (weiter unten beschrieben).

Ein Decompiler ist ohne Zweifel eine nützliche Hilfe bei der Programmentwicklung und wird wohl auch in einer der nächsten Ausgaben der VIERTE DIMENSION als Sourcelisting vorgestellt werden. Vorab, hier nun eine kurze Decompilations-Session mit dem Decompiler von Ray Duncan für Z80-FORTH. Kopien des Sourcelistings (1 Seite) für diesen Decompiler sind beim C.I.A (Red.) für 15 Pfennige pro Seite + Porto zu haben.

DIS LIST

```
1F29 :
1F2B ?DECIMAL
1F2D CR
1F2F DUP
1F31 SCR
1F33 !
1F35 Prnt Text: Screen #
1F41 .
1F43 16 (10 H)
1F47 0 (0 H)
1F4B (DO)
1F4D CR
1F4F R
1F51 3 (3 H)
1F55 .R
1F57 SPACE
1F59 R
1F5B SCR
1F5D $ ( fetch -- Anm.d.Red )
1F5F .LINE
1F61 LOOP to 1F4D
1F65 CR
1F67 ;S
ok
```

PAST APPROACH TO SYSTEM DESIGN AND DEVELOPMENT



BUCHBESPRECHUNG

Zech, R.: Die Programmiersprache FORTH. Franzis-Verlag, München 1984, ISBN 3-7723-7261-9, 312 S., DM 68.-

Der Untertitel dieses brandneuen Buches lautet "Eine kompakte Darstellung der Programmiersprache mit vielen Beispielen", und es gliedert sich in folgende Hauptabschnitte: Aufbau des FORTH, Programmieren in FORTH, Programmstrukturen und strukturierte Programmierung, das Vocabulary, FORTH für Fortgeschrittene, FORTH I/O und die Installation auf konkreten Systemen. Es folgen 11 Anhänge, ein Literaturverzeichnis sowie ein knappes Sachverzeichnis. Vestreut im Text finden wir 47 schematische Abbildungen, die vor allem Speicherbelegung und Programmablaufpläne verdeutlichen, sowie 36 Tafeln vorwiegend mit der Erklärung von Definitionen wichtige FORTH-Worte.

Für die Programmiersprache FORTH, speziell FIG-FORTH, konnten bisher im wesentlichen nur englische Bücher empfohlen werden. Sehr beliebt ist von Leo Brodie das "Starting FORTH" und als Orientierungsbuch "FORTH User's Manual" (Rockwell-Buch für das AIM-65 FORTH). Das von Zech verfasste Buch ist anders als alle anderen. Es erhebt zwar den Anspruch auch ein Anfängerbuch zu sein, in Wirklichkeit schaut es der Sprache von Anfang an sehr "unter die Haube" (Zitat nach Brodie). Nach dem Urteil des Rezensenten, der selbst viel zu FORTH geschrieben hat, ist das Buch mehr ein Werk für Fortgeschrittene und Systemprogrammierer. Wenn man es mit dieser Erwartung aufschlägt, dann findet man wirklich handfeste Informationen auch zu schwierigen Gegenständen der Sprache. Das

Buch besticht die vollständige Durchdringung aller Aspekte von FORTH. Es ersetzt damit eine ganze Reihe englischer Texte und läßt die bis jetzt verfügbaren deutschen Texte weit hinter sich. Von seinem Umfang und seiner Zielsetzung ist es mehr für Profis und solche die es werden wollen. Für Anfänger sei hier nochmal erwähnt daß es das beliebte Buch "Starting Forth" von Leo Brodie nun auch auf deutsch gibt.

"Programmieren in Forth"
Hanser Verlag
8000 München 80
Kolbergerstrasse 22
Tel.: 089/98 25 11

5. DIMENSION

Liebe Freunde !

Als der Initiator der Arbeitsgruppe FÜNFTE DIMENSION der FORTH-Gesellschaft Deutschland grüße ich Sie herzlich. Erlauben Sie mir, daß ich mich vorstelle. Mein Name ist Andreas Goppold. Ich bin der Metaprogrammierer der FORTH-Gesellschaft. Die Assoziation des Wortes Meta beinhaltet, wie in dem Wort Metaphysik, etwas, das hinter den Dingen liegt. Das beweist auch der Titel dieser Publikation: die FÜNFTE DIMENSION kommt konsequenterweise hinter der Vierten. (Meta heißt: Dahinter kommend) Ich sehe es als meine Hauptaufgabe, mich mit Dingen zu beschäftigen, die über Anwendung von FORTH als nützliches Werkzeug, mit dem man Probleme schnell lösen kann, hinausgehen. Ich sehe in FORTH nicht nur das, sondern auch eine Prototypsprache für Interaktives Programmieren. Was ich damit meine, ist in meinem Artikel "Das Paradigma der Interaktiven Programmierung" und zum Teil auch in "FORTH - Das Programmiersystem ohne Grenzen" enthalten. Beide Artikel sind zu umfangreich, als daß sie in den Rahmen dieses Newsletters passen würden, und sind deshalb separat von der FIG erhältlich.

Für mich und meine Mitstreiter in der FÜNFTEN DIMENSION ist FORTH also nicht allein eine Programmiersprache, sondern die Beschäftigung mit FORTH ist Teil eines geistigen Abenteurers, einer Entdeckungsreise an die Grenzen des Denkbaren. FORTH überschreitet so viele Grenzen herkömmlicher formaler Systeme, daß wir glauben, daß unsere Arbeit einen gewichtigen Beitrag zu der Entwicklung der Computerei liefern kann, und vielleicht noch ein bißchen darüber hinaus. Wir sind also nicht ganz unbescheiden in unseren Ambitionen, und ich glaube mit Recht. Unser kleines Problem ist, daß es ungeheuer viel Neuland gibt, das zu bearbeiten ist, und wir sind jetzt noch sehr wenige. Aber wir sehen einen sehr konsequenten Ausweg aus dieser Zwangslage. Wir sehen die Chance des SELBST-BOOTSTRAP. Was das ist, möchte ich im Augenblick nur mit einem Beispiel deutlich machen: Charles Moore, der Erfinder von FORTH, hat sich als den ersten "Computer-unterstützten Programmier" bezeichnet. Ich habe noch einiges dazu in meinem Artikel: "Interaktives Programmieren" gesagt. FORTH ist ein Werkzeug, das mit seinem Benutzer wachsen

kann, die Co-Evolution von Mensch und Maschine. Wir sehen also noch einige andere Perspektiven in der Entwicklung der Technologie, nicht nur Mega-Maschinen und Megalocorporations sondern Intelligenzverstärkung und noch einiges, was in diesen Bereich fällt. Wir sehen auch einen gangbaren Weg: Man nehme einen Home-Computer, man nehme FORTH, und man fange an zu programmieren. Dann teile man seine Arbeitsergebnisse in dem Forum der FORTH-Gesellschaft mit, und aus dem allgemeinen feedback resultiert weiterer Fortschritt.

Ich möchte nur noch ein paar der Punkte anreißen, die ich in der Arbeitsgruppe behandeln will. Eins der Ausgangsthemen wird Hofstadters Buch: "Gödel, Escher, Bach" (GEB) sein, das der Klett/Cotta Verlag dankenswerterweise nun auch ins "Computerentwicklungsland" gebracht hat. Ich glaube nicht, daß Hofstadter hier so ein Bestseller werden wird, wie in den USA. Aber vielleicht können wir ein wenig zur Weiterentwicklung dieser Ideen beitragen. Nicht zu vergessen, zwei seiner Titelpersonen waren ja aus Deutschland. Wir haben hier noch etwas zu erledigen.

Was ist die Verbindung von Hofstadters Buch und FORTH ? Nun, ersteinmal sind die Meisterwerke von Bach auf dem selben Prinzip aufgebaut, wie FORTH Stacks. Die Musik von Bach kann am besten verstanden werden, wenn man sie als die Interaktion mehrerer Stacks betrachtet. Weiterhin redet Hofstadter von Selbstreferenz. Dieses Prinzip ist die nächste Entsprechung der Funktionsweise von FORTH-Definitionen. Selbstreferenz wird sowohl in der Natur, als auch in der Kunst oft verwendet. Alles baut sich auf einen Satz von universell verwendbaren Bausteinen auf, in der Natur sind das die Aminosäuren, in FORTH die Primitives. Diese werden zu Komplexen zusammengefaßt, die wiederum Bausteine für noch größere Komplexe sind. Zwischen diesen Komplexen besteht eine hierarchische Strukturierung. Die Biologie unterscheidet säuberlich nach dem molekularen Level, und dem zellularen Level. In FORTH sind das eben die Worte und die Vokabulare. Man ist noch nicht soweit in der Programmierung, daß man Komplexitäten, die auch nur dem molekularen Level der Biologie entsprechen würden, handhaben könnte. Über dem zellularen Level gibt es aber noch das der Organstrukturen, und das der Lebensräume, und

letztendlich das der Erde als zusammenhängendes System. Die Natur bewältigt die Organisation sozusagen "natürlich". Wie das gemacht wird, kann mit dem Begriff "Selbst-Organisierende Systeme" umrissen werden, (wenn es auch augenblicklich damit nicht gerade erklärt wird).

Wir können also sehen, welchen Weg die Wissenschaft noch vor sich hat. Jede weitere Ebene der Organisation ist etwa hundertmal oder tausendmal komplexer als die darunter. Heute denkt man schon an organische Computer, aber mein Eindruck ist, daß die Programmiermethoden noch nicht einmal die Ebene von Megabyte Memories erreicht haben.

Die Software Gilde

Außer der Arbeit, die nahe bei der Programmierung liegt, haben wir noch andere Dinge vor: Die Organisationsprinzipien, die in FORTH möglich sind, können auch auf andere Bereiche angewandt werden. Warum muß man einen Computer haben, um das Prinzip anzuwenden ? Das Gehirn ist doch immer noch der beste Computer. Wir werden uns hier damit beschäftigen, wie wir unseren Gehirn-Computer optimieren, ob wir nun damit besser programmieren wollen, oder etwas anderes. Bootstrap fängt immer bei einem selbst an, da wo man steht. Das nennen wir dann die Selbs-Meta-Programmierung. Ja, der Platz hier auf der Seite, den mir die FORTH-Gesellschaft zur Verfügung gestellt hat, wird langsam knapp. Ich habe in meinen Artikeln einiges mehr zu diesen Themen gesagt, und wenn Sie meine Artikel lesen wollen, oder in irgendeiner Weise an der Arbeitsgruppe mitmachen wollen, setzen Sie sich bitte mit mir in Verbindung.

Andreas Goppold
Arbeitsgruppe
FÜNFTE DIMENSION

FORTH GESELLSCHAFT
Bethesdastraße 11 c
2000 Hamburg 26
tel.: 040-254556

Die Artikel:

"FORTH - ein Programmiersystem ohne Grenzen"
und
"Das Paradigma der Interaktiven Programmierung"

sind bei der FORTH GESELLSCHAFT zum Preis von DM 5.- zu erhalten

(Forts. v. Seite 3)

einen Hauptverantwortlichen und einen Ersatzmann. Wir hatten den 8080, 6800, PDP-11, Pace, 1802 und TI9900 mit jeweils zwei Leuten - wie bei Noah und der Arche: jeweils zwei für jeden Prozessortyp. Dann begann, was heute zu den Samstags-FIG-Treffen in Nord Kalifornien geworden ist. Kim Harris war unser Schriftführer und machte akribische Protokolle über den Fortgang der Arbeit.

Wir erläuterten den Implementatoren das System schrittweise, und wenn sie Probleme entdeckten, dann tauschten wir Fragen und Analysen mit den anderen Gruppen aus. Wir suchten nach Maschinentransportablen Lösungen. Daraus entwickelten sich zwei Verbesserungen. Dave Kilbridge arbeitete an der Arithmetik, da in der ersten Implementation einige Vorzeichen und Überträge nicht ganz stimmten. Inzwischen hatten wir Kontakt mit den Leuten der Universität in Utrecht in Holland. Sie tauschten mit uns den gesamten Quelltext ihres Systems aus, so ungefähr 2000 Screens und gestatteten uns, Teile daraus in unserer Implementation zu verwenden. Wir übernahmen die gesamte Fehlererkennung im Compiler aus ihrem Code; bis dahin hatte keines der Systeme in den USA etwas derartiges.

Für dreizehn der Mitglieder im Implementationsteam war das erste Forth System, das sie je sahen, dasjenige, das sie selber entwickelt hatten. Wir hatten einige wenige grundlegende Regeln aufgestellt, die von allen befolgt wurden und das bewährte sich prächtig. Sie folgten im Grunde dem Modell und versuchten so wenig wie möglich davon abzuweichen, da es uns klar war, daß dies ein Ausgangspunkt sein würde, in dem die gleiche Information in einer Vielzahl unterschiedlicher Formen vorhanden sein sollte. Wir trieben es sogar so weit, die gleichen Label im Assembler Quelltext zu benutzen; auf einer unserer letzten Sitzungen legten wir diese Namen fest, und sie hatten in der gleichen Reihenfolge benutzt zu werden usw.

Die Universität von Utrecht hatte Forth also schon so früh?

Ja, es war aus Chuck's ursprünglicher Arbeit in Kitt Peak entstanden. Astronomen aus der ganzen Welt arbeiteten für jeweils einige Zeit in Kitt Peak. In Abhängigkeit von der Hardware, die sie zu Hause hatten, nahmen sie ein Band mit nach Hause und - zack - sie hatten Forth. So lief also

Forth in St. Andrews in Schottland, an einigen Stellen in Frankreich, in Chile und Cal-Tech. Das war alles vor der Zeit von Forth Inc., so zwischen 1969 und '73. Forth begann seine Migration in der Welt der Astronomen.

Damals war Forth in Wirklichkeit ein Betriebssystem für, wie ich sage, verkrüppelte Rechner. Forth wurde sehr positiv von Nutzern von Rechnern mit sehr begrenzten Möglichkeiten, was den Hauptspeicher, den Massenspeicher und Ein/Ausgabe angeht, angenommen. Der VARIAN 620i war ein Krüppelrechner. Einige der frühen Hewlett-Packard Rechner, der 2100 zum Beispiel, hatten nur sehr begrenzte Unterstützung durch den Hersteller. In diesen Fällen wurde Forth sehr geschätzt.

Andererseits haben wir in der Forth Gemeinde ein sehr reales Problem; dadurch, daß Hersteller bessere Software anbieten, verschwindet die Notwendigkeit und Nachfrage nach Forth. Vor fünf Jahren bot Forth unersetzliche Möglichkeiten. Nun scheint es, als ob einige dieser Möglichkeiten nicht mehr so attraktiv sind. Zum Beispiel ist heute mehr Hauptspeicher vorhanden. Ein/Ausgabe ist schneller, es gibt mehr Plattenplatz, und die Dateistrukturen sind weniger einengend. Dies alles fordert Leute, die Forth Systeme benutzen oder schreiben heraus. Werden sie weiterhin in der "Krüppelkomputer" Mentalität verharren oder werden sie wachsen und den Anforderungen genügen?

Wenn man die Eigenschaften von Forth und den heutigen Computern in Betracht zieht - ist die Sprache noch zeitgemäß?

Forth vereinigt in sich die richtigen Elemente aber ich fürchte, daß diese im allgemeinen weit weniger genutzt werden als sie sollten. Wir kommen am Ende eines Fünf-Jahres-Zyklus an einen Punkt, den Hans Niewenhuijzen im Jahre 1979 vorhergesagt hat. Forth selber war ein Fundament, eine Basis, auf der höhere Sprachkonstrukte entwickelt werden würden. Hans hat diesen Gedanken auf Konferenzen verbreitet, aber die Botschaft wurde nicht recht gehört. Hans hat gesagt: "Schaut euch die anderen Sprachen wie Algol, Pascal und Lisp an; nehmt die Komponenten, die gut sind, integriert sie in Forth und macht eine höhere Programmiersprache aus Forth". Hier konnte dieser Gedanke niemals richtig Wurzeln schlagen, und Hans war darauf-

hin enttäuscht. Er hatte das Gefühl, daß seine Ideen nicht anerkannt würden. Ich denke, daß es in Wirklichkeit so war, daß wir die Unzulänglichkeiten von Forth nicht erkannten.

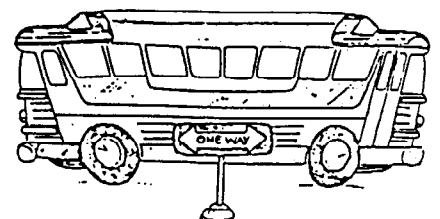
Die letzten fünf Jahre haben mir deutlich werden lassen, daß die Unzulänglichkeiten nicht in der Sprache selbst liegen, sondern in Beziehung auf die Dienstprogramme. Zum Beispiel das Fehlen eines Dateisystems und die begrenzten Möglichkeiten Informationsmengen zu manipulieren, die größer sind als eine Zahl auf dem Stack oder einem Characterstring. Einige andere Programmierumgebungen wie z.B. UNIX gestatten die Manipulation von Dateien, oder das Pipen bzw. Übertragen von Information von einer Datei in eine andere, während sie auf dem Wege aus der einen Datei heraus in die andere hinein manipuliert werden kann. Diese höheren Programmierkonzepte sind nicht in Forth integriert worden; kurz: Forth stagniert seit vier oder fünf Jahren. Die Verbesserungen in Forth sind sehr gering verglichen mit denen in anderen Programmierumgebungen.

Was mich beunruhigt ist, daß wir langsamer werden, und daß Benutzer von Forth heute nicht viel mehr damit machen können als vor vier oder fünf Jahren. Die zentrale Frage ist, wie wir die Einstellungen, das Interesse und die Umgebung schaffen, so daß immer mehr Leute auf Forth aufbauen und es dadurch immer interaktiver machen, so daß es immer nützlicher wird und abstraktere Datentypen manipuliert als nur Zahlen und ein paar Zeichen.

(Forts. Seite 9)

AUS DEM FORTH-DICTIONARY:

-TRAILING



EIN GANZ EINFACHER FALL

Forth hat in seinem Kern kein CASE Statement. Dies etwa nicht deshalb, weil es besonders schwierig wäre, oder weil man es vergessen hätte; nein, der Hauptgrund, warum es nicht im Kern enthalten ist, besteht darin, daß es einfach zu selten benötigt wird, so daß es sinnlos ist, es immer "mitzuschleppen" - ganz abgesehen davon, daß es grundsätzlich drei verschiedene Arten von CASE Statements gibt und deshalb nach den murphischen Gesetzen im gegebenen Fall immer der falsche Typus zur Hand wäre.

Im folgenden werde ich die simpelste Art eines CASE Statements vorstellen, die ich kenne; die in allen den Fällen eingesetzt werden kann, wenn z.B. in Abhängigkeit von einzelnen Keys von der Tastatur einige wenige, unterschiedliche Aktionen ausgelöst werden sollen, wie dies z.B. in dem Forth Wort EXPECT notwendig ist - dort nämlich die Spezialbehandlung von <space>, <backspace> & <return>

Zunächst aber etwas Theorie, nämlich die drei verschiedenen Typen von CASE Statements:

1. "positional" case

Im Falle "0" soll eine, im Falle "1" eine Andere, im Falle "2" noch eine Andere, usf. Aktion ausgelöst werden. D.h., die einzelnen Fälle sind fortlaufend "durchnummeriert". Darauf will ich heute nicht besonders eingehen, obwohl die ser Fall in Forth einfach und effizient zu lösen ist. Aber Merke: Der erste Fall ist Fall "0", wie überhaupt in Forth grundsätzlich ab "Null" gezählt wird - wer das nicht beherzigt, der zahlt dadurch, daß er diverse 1- und 1+ in seinem Code versprenkeln muß.

2. "keyed" case

Im Falle "12" soll eine, im Falle "534" eine andere usf. Aktion ausgelöst werden, und wenn keiner der aufgeführten Fälle zutrifft, dann irgendeine Standardaktion. Es werden unterschiedliche Code Sequenzen ausgeführt in Abhängig einer Zahl, die mit vorgegebenen, gestreuten Werten verglichen wird. Eine solche Struktur kann in einfacher Weise mit dem Wort case? und den üblichen Kontrollstrukturen implementiert werden. Davon später mehr.

3. "range" case

Im Falle "13" bis "20" ("13-20") dies, bei "21-39" das, etc. und ansonsten. Selten gebraucht.

Als Motivation nun zunächst ein kleines Spiel, aus dem die Benutzung von case? deutlich werden sollte

```
013 CONSTANT <CR>
```

```
: wastun ( Zahl - Flag)
  ASCII j case? IF ." ja"      False EXIT THEN
  ASCII n case? IF ." nein"   False EXIT THEN
  <CR> case? IF ." fertig" True EXIT THEN
  emit ." ungültig" False ;
```

```
: Spiel BEGIN KEY wastun UNTIL ;
```

Ich hoffe, daß es ziemlich offensichtlich ist, was passiert; wers nicht durchschaut, muß halt ausprobieren, daß ihn nur ein <cr> aus dem Spiel befreit.

Ungewohnt mag die Benutzung von EXIT erscheinen (in FIG Systemen auch ;S genannt). Mit EXIT wird wastun verlassen und als nächstes wird dann wieder

ein logischer Wert UNTIL zum Fraß vorgeworfen. Allgemein kann EXIT benutzt werden, um ELSE Verzweigungen zu vermeiden. Merke: Jeder ELSE Zweig, der nicht vorkommt, braucht auch nicht durchdacht zu werden, wenn man sich später einmal das Programm anschaut, oder wenn etwa jemand Anderer es verstehen soll. Wers nicht einsieht, der erinnere sich einmal, welche Editierbemühungen es kostet, eine verschachtelte Struktur deutlich zu machen - viel einfacher ist das Denken in geschichteten Strukturen. Psychologische Untersuchungen legen den Schluß nahe, daß der Mensch mit fünf bis sieben abstrakten Objekten "jonglieren" kann. Das bestätigt die gefühlsmäßige Erfahrung, daß sich gutes Forth durch viele kurze Definitionen auszeichnet, die aufeinander aufbauen - die Alternative ist "Spaghetticode", Nährboden zahlreicher "Prettyprinter" und anderer Analyseprogramme.

Was False und True im Programm zu suchen haben? - Das sind simple Konstanten.

```
0 CONSTANT False      1 CONSTANT True
```

ähem, ab dem 83er Standard ist True allerdings -1 !

Bevor ich nun die Definition von case? hin-schreibe, zunächst einmal einige inzwischen weitverbreitete Worte, um den Stack zu manipulieren; die werden so häufig gebraucht, daß es sich lohnt, sie mit in den Kern zu packen.

```
: nip ( n0 n1 - n1)      swap drop ;
: under ( n0 n1 - n1 n0 n1) swap over ;
: stash ( n0 n1 - n0 n0 n1) over swap ;
```

Nach diesen ganzen Vorbereitungen denn nun:

```
: case? ( n0 n1 - tf / n0 ff)
  over = dup IF nip THEN ;
```

Zwei Zahlen liegen auf dem Stack, n0, meine "Eingabe", die einen Fall auswählen soll und n1, ein Vergleichswert. Stimmen n0 und n1 überein, dann habe ich meinen gesuchten Fall gefunden und case? kommt einfach mit tf, einer "True Flag" zurück, sodaß das im allgemeinen folgende IF passiert werden kann; stimmen n0 und n1 nicht überein, nun, dann wird n0 für weitere Vergleiche aufbewahrt und ff, eine "False Flag" ermöglicht das Überspringen des folgenden IF's.

(c) 1984, K. Schleisiek

Und hier nun eine konkrete Anwendung des CASE? :

```
: Arbeitsgruppen ( n - )
  FORTH-83 CASE?
  IF customize distribute exit THEN
LEIBNIZ CASE?
  IF Syntax definieren exit THEN
5th-Dimension CASE?
  IF rumspinnen exit THEN
Newsletter CASE?
  IF BEGIN Artikelte write AGAIN THEN
Dokumentation CASE?
  IF BEGIN Info's write AGAIN THEN
PASCAL > IF AG gründen exit THEN
  0 Interest !
: FIGEN BEGIN Interest $ DUP Basic >
  WHILE Arbeitsgruppen REPEAT ;
```


Meinst du, daß einige unserer Standardisierungsbemühungen überdacht werden sollten, um kreativere Herangehensweisen zu ermöglichen?

Keineswegs. Nein, ich glaube, daß die Arbeit des Forth Standard Teams, wie sie zur Zeit stattfindet, genau die richtige Aktivität auf der richtigen Ebene ist. Sie ist weder auf einer zu tiefen Ebene - nur Codeworte und einige Primitives; noch auf einer zu hohen - wie z.B. zu versuchen, einen Editor oder ein Telekommunikationsprotokoll zu standardisieren. Ich glaube, daß es genau richtig ist. Der Standard ist umfassend genug um nützliche Anwendungen zu schreiben, die sich hauptsächlich an den Standard halten, obwohl kleinere Teile möglicherweise so speziell sind, daß diese Nicht-Standard sein müssen.

Wir sind heute bereit für eine neue Ebene von Bemühungen und Abstimmungen zwischen Benutzern und Anbietern auf Gebieten wie einer Dateistruktur und einem Zuteilungsmechanismus für den Massenspeicher. Damit meine ich Dateien, die mit einer begrenzten Menge von Regeln assoziiert sind. Nicht "keine Regeln", aber mit einer kleinen und einfachen Menge von Regeln, so daß ich in der Lage bin, Datenstrukturen variabler Länge zu erzeugen, die Dateien genannt werden. Andererseits sollte ich nach wie vor in der Lage sein, Daten in blocks zu verwalten, wenn ich dies wünsche. Das heißt, daß ich das System beauftrage, eine Datei für mich anzulegen, in das ich Daten ablegen kann; oder ich beauftrage das System, mir einige blocks zu geben, da für bestimmte Anwendungen blocks besser geeignet sein können. Ich möchte zusammenhängende oder verteilte Blöcke beantragen können; und wenn ich fertig bin, möchte ich sie an das System zurückgeben und "vielen Dank" sagen können. Dies scheint mir ein minimaler nächster Schritt.

Darüber hinaus beginnen wir die Benutzerschnittstelle zu betrachten und wie diese flexibler gemacht werden kann für Sachen wie Graphik, Klänge und Manipulatoren wie Lichtzeiger, Maus oder Graphiktablett. Wie werden diese angesteuert in einer Forth Umgebung, so daß wir nicht durch eine bestimmte Hardware eingengt werden?

Werden sich diese Dinge entwickeln oder ist dafür eine Sprache der fünften Generation notwendig?

Nein, nein, nein. Keine neuen Durchbrüche! Alles in handhabbaren, verdaulichen, modularen Stücken. Keine neue fünfte Sprache, nicht das Kind mit dem Bade ausschütten. Dennoch glaube ich, daß wir alle gut daran täten, uns zu fragen, wie eine Programmierumgebung aussehen sollte, und Weiterentwicklungen zu unterstützen. Wird es durch heutige Anbieterkanäle, durch akademische Unterstützung, durch Benutzergruppen oder allein durch innovative Individuen, die auf sich selbst gestellt arbeiten, kommen? Die Straße, auf der diese Erweiterungen kommen werden, ist mir keineswegs klar.

Es besteht hier ein weites Gebiet von Möglichkeiten für die Softwareanbieter. Ein interessantes Individuum, auf das man achten sollte, ist Pierre Moreton, der zur Zeit in Palo Alto arbeitet. Er wird ein portierbares Dateisystem veröffentlichen, das er im Laufe mehrerer Jahre entwickelt hat. Ich habe von Benutzern gehört, daß es sehr gut ist. Ich will hier nicht Pierre's Dateistruktur als Standard vorschlagen; was ich sagen will, ist, daß andere sein Werk gesehen und für gut befunden haben, es anerkennen und benutzen. Das zeigt ihren Wert. Beispiele dieser Art von Weiterentwicklung sind ziemlich selten; d.h. es gibt davon bisher sehr wenige, und es bestehen ungeheuer viele Möglichkeiten für weitere. Ren Curry's Programmierwerkzeuge werden von vielen gerühmt, die ohne diese Art von Debugger, Breakpointer und Decompiler, keine funktionierenden Systeme hätten.

Die Anbieter bewegen sich zur Zeit auf einem schmalen Pfad zwischen gutem Forth und Beiträgen zu zukünftigen Produkten. Für meinen Geschmack sind sie etwas zu konservativ. Bisher versuchen die Anbieter lediglich, Forth sauber zu implementieren und zu dokumentieren; sie ziehen sich zu früh zurück. Sie stellen sich nicht den Forderungen, bearbeiten noch nicht unrealisierte aber notwendige Aufgaben. Ich kenne kein einziges kommerzielles Spreadsheet Programm in Forth. Es gibt QTF, das Textprogramm von Leo Brodie; das arbeitet aber noch auf der Blockebene und ist ein sehr beschränktes Textsystem. Es ist gut genug, um die Arbeit zu tun, aber man würde es nicht benutzen, wenn es etwas Besseres gäbe. Ich glaube, daß

hier eine große Gelegenheit für jemanden besteht, ein gutes Forth System als Basis und darauf die Programme anzubieten, die man heute einfach von einem Computer verlangt.

Was ermöglicht es dir, dich in Situationen wie Treffen von ehrenamtlichen FIG Mitarbeitern zu begeben, in denen emotionelle Spannungen herrschen und dennoch mit einem Konsens herauszukommen, den alle mittragen können? Es gibt Streit, und es wird debattiert, es wird abgestimmt und ein Beschluß gefaßt und man arbeitet weiterhin zusammen.

Es mag ein bißchen abgedroschen klingen, aber die Motivation durch und das gemeinsame Interesse an Forth transzendiert eine Menge solcher kurzzeitigen persönlichen Prioritäten. Ich will keinesfalls darauf anspielen, daß es sich um spezielle Führungstechniken oder um eine einzigartige Umgebung handelt, durch die das passiert. Es hat allein etwas mit der Begeisterung der Individuen zu tun; es findet natürlich ein gewisser Ausleseprozeß dadurch statt, daß Leute, die total andere Ansichten haben, nach einiger Zeit nicht mehr teilnehmen. Deshalb sind wir eine ziemlich konsistente Gruppe, die zumindest in der Lage ist, andere Ansichten wahrzunehmen.

Der Aspekt der Ehrenamtlichkeit ist ein Hauptelement; dadurch wird jedem klar, daß er für seine eigenen Interessen eintreten muß, um die Teilnahme zu genießen. Wir müssen das anerkennen und uns darauf einstellen. Es geschieht, weil sie es wollen und weil es notwendig ist, und nicht, weil jemand es fordert. Sie tun es auch, weil ein Abhängigkeitsgefühl existiert: was auch immer jemand tut, es gibt andere, die davon abhängig sind, daß sie es tun. Zum Beispiel die Gruppe, die die FORML Konferenz organisiert. Sie tut dies Jahr für Jahr, weil sie die Notwendigkeit für ein solches Forum und die Herausgabe der Proceedings sieht, und daß ein breites Spektrum von Meinungen auf der Konferenz präsentierte ist. Es ist das gemeinsame Verlangen und das Gemeinwohl, das es zum Blühen bringt.

Ich sehe uns an einem bedeutenden Wendepunkt in der Entwicklung von FIG. Bis heute war es notwendiger- und korrekterweise in Nord Kalifornien zentriert, weil es hier genug Leute gab, um die kritische Masse zu erreichen, die notwendig war, um die Arbeit zu erledigen. Jetzt sind wir an einem Punkt angekommen, an

dem wir die Teilnahme unter den Mitgliedern aufteilen können, so daß FIG als eine Organisation gestärkt wird, die durch ihre Mitglieder lebt - eine Organisation, die kontrolliert und geführt wird durch und abhängig ist von ihren Mitgliedern. Bis heute ist die Organisation so, daß die zentrale FIG Planungsgruppe die Mitglieder bedient: Veröffentlichungen, Quelltexte, Konferenzberichte und andere Ereignisse. Ich glaube, daß das Pendel jetzt dahin schwingt, daß die Arbeit, wie zum Beispiel die Entwicklung der Chapter durch John Hall, anfängt Früchte zu tragen. Ich hoffe, daß wir in den nächsten Jahren unsere Jahrestreffen regionalisieren, so daß es mal in Washington, Dallas und New Orleans sein wird. Wir werden eine effektive Mitgliederorganisation daran erkennen, daß wir uns zu verschiedenen Orten bewegen können, wobei die Organisation lokal bewältigt wird.

Warum steckts du soviel freiwillige Arbeit in FIG? Manche Leute sind überrascht, daß du bereits seit sieben Jahren aktiv bist.

Das höre ich nicht zum erstenmal. Einige mißtrauische Zeitgenossen glauben, daß hinter einem mehrjährigen unbezahlten Einsatz irgendein finsterner Grund stecken muß. Die offensichtliche Antwort darauf - jedenfalls für mich - ist, daß ich einfach Spaß an sowohl mündlicher wie auch schriftlicher Kommunikation habe. Um über Forth zu sprechen, brauche ich Gesprächspartner; also ist für uns alle eine wachsende Gruppe wichtig um ein Forum zu haben.

Wie schätzt du die Rolle von FIG heute ein?

Es wird noch immer sehr stark nach der Antwort auf die Frage gesucht, welche Dienstleistungen FIG bieten soll und welche Veränderungen in diesem Aspekt stattfinden. Zu einer sehr frühen Zeit sollte FIG einen Newsletter herausgeben und Leute ausbilden, um solche Ziele zu erreichen wie, wie Leute an Forth herangeführt werden können und wie diese Forth produktiv nutzen können, um ihre Probleme zu lösen.

Eine Frage, die heute breit debattiert wird ist, ob FIG einen bedeutenden Teil ihrer Mittel dazu verwenden soll, Forth zu popularisieren, d.h. an Leute heranzutragen, die sonst nicht damit in Berührung kämen. Oder sollte ihre Rolle hauptsächlich darin bestehen, die Mitglieder zu unterstützen - Leute, die

Forth bereits kennen und ihr eigenes Verständnis und ihre Fähigkeiten erweitern wollen? Kurz: Haben wir die Public-Relation Aufgabe, Forth zu legitimieren und Leute auf es hinzuweisen, die sonst keinen Kontakt damit hätten? Einige meinen, daß dies eher Aufgabe der Anbieter wäre; daß FIG erst in dem Augenblick eine Rolle spielen sollte, wenn jemand Forth bemerkt hat und sein Verständnis erweitern möchte. Bisher gibt es dazu noch keine endgültigen Antworten und je mehr Ansichten wir zu diesem Thema hören, desto besser.

Warum ist FIG noch immer nützlich?

Darauf gibt es eine einfache Antwort. Es gibt nichts anderes. Es ist die einzige Interessengemeinschaft, die sich um Forth entwickelt hat. Das Standard Team ist eine Gruppe von nur 26 Individuen und hat Perioden hektischer Aktivität zwischen denen lange Ruheperioden liegen. Ein Hersteller hat seine eigene Benutzergruppe gegründet. Ich kenne nicht ihre Struktur aber nach allen Anzeichen, die ich sehen kann, spielt sich nicht sehr viel ab. Aber FIG, wie gesagt, gibts. Hört sich das hämisch an?

Der Aspekt, daß FIG eine Mitgliederorganisation ist, wird erst jetzt so langsam bewußt und entwickelt. Ich habe das sichere Gefühl, daß wir in den nächsten zwei, drei Jahren erhebliche Veränderungen erleben werden in der Zusammensetzung von FIG, und wie sie mit den Mitgliedern zusammenarbeitet und von ihnen abhängig ist. Einige Neuerungen werden dieses Jahr eingeführt: Mitgliederausweise, Discountabmachungen und möglicherweise telefonische Beratung für Mitglieder. Es gibt wohl ein halbes Dutzend Aspekte, die wir für die Mitglieder entwickeln wollen. Es ist ein typischer Fall von zwei plus zwei ist fünf - wenn mehr Menschen zusammenkommen, dann bekommen sie bessere Resultate aus ihren gemeinsamen Interessen.

Wird FIG die F83-Implementation von H. Laxen und M. Perry als Sprachmodell empfehlen?

Ich hoffe doch. Ich glaube, daß das Standards Team etwas versäumt hat: Nachdem die Mitglieder ihre Zustimmung zum Entwurf gegeben hatten, glaubten sie, ihre Arbeit sei getan. Es wird mir nun klar, daß das nicht der Fall war. Es fehlt noch etwas, nämlich die tatsächliche Annahme des Stan-

dards durch eine Gruppe von Benutzern. Ein Standard ist solange kein richtiger Standard, bis er breit akzeptiert ist und benutzt wird. Eines der Ziele von FIG ist es, formell aktiv zu werden im Namen der Mitglieder in Beziehung auf den Standard - eine Überprüfungsprozeß einzuleiten, um ihn auf unsere gemeinsamen Anforderungen abzuklopfen. Wenn ein solches Prüfungsgremium es für richtig befindet, würde es im Namen der Mitglieder und von FIG mit einer Empfehlung herauskommen, daß der Standard benutzt werden sollte.

Was für ein Verhältnis hast du zu den anderen Gruppen wie zum Beispiel Forth Inc. und der Anbietergruppe?

1979 habe ich Forth Inc. Aktien gekauft und bin daraufhin in ihre Gesellschafterversammlung aufgenommen worden. Dort habe ich von 1980 bis Ende '82 als Gesellschafter gewirkt und dann bis Oktober '83 als Vorsitzender. Zur Zeit ist mein Engagement nur noch das eines Investors.

Mitglied in der Anbietergruppe bin ich seit der Gründung vor zwei Jahren. Mein Produkt ist der Ultra Compiler, ein Target Compiler für industrielle Kunden. Ein Anbieter verkauft es außerdem z.Z. als Teil eines Gesamtsystems weiter. Ganz sicher werde ich mich auch in Zukunft mit Forth Anwendungspaketen beschäftigen.

Gibt es einen Markt für Forth und Forth-Produkte?

Es gibt einen Markt für Forth Programmierer, aber noch nicht für Forth Produkte. Bis heute gibt es genug industrielle Anwendungen von Forth, so daß professionelle Forth Programmierer anscheinend keine Schwierigkeit haben, Arbeit zu finden. Ich weiß von zwei Firmen, die im vergangenen Jahr überlegt haben, Forth als Programmiersprache fallenzulassen, weil sie keine Leute fanden. Ein Projekt war von jemandem mit Forth begonnen worden, der sich damit auskannte, der hatte dann die Firma verlassen und die hatten dann dermaßen große Schwierigkeiten, andere kompetente Leute für das Projekt zu finden, daß sie nachdrücklich darauf beharrten, von Forth wegzukommen.

Die Fähigkeiten von Forth Programmierern sind verkäuflich. Andererseits ist es mir keineswegs klar, daß ein Markt für Forth selber existiert; bei speziellen Anwendungen, die auf Forth aufbauen, sieht das dann wieder besser aus. Ich möchte das dadurch unter-

mauern, daß man sich mal Firmen wie Digital Research und Microsoft und ihre Verkaufspalette anschaut. Sie liefern Sprachen und Betriebssysteme, im grunde genommen Werkzeuge für Programmierer. Nun, das ist genau das, was Forth ist. Nehmen wir mal die zehn größten Firmen in der Forth Umgebung - jeder möge sich selber überlegen welche das sind - und addieren ihren jährlichen Umsatz, dann kommen wir maximal auf ca. drei Millionen Dollar, oder realistischer auf zwei Millionen. Digital Research und Microsoft sind 6000 % größer. Um es andersherum zu sagen: Die Forth Community muß um das Hundertfache wachsen; es existiert eine hundertfache Wachstumsmöglichkeit für die Forth Community, um sich einen eigenen Markt zu erschließen.

Wenn es nun wahr ist, was wir immer sagen, daß Forth super ist, so produktiv, so innovativ, so nützlich und so viele Probleme löst - warum gibt es dann diese einhundert zu eins Schere zwischen Forth und den anderen Programmierumgebungen? Die Herausforderung liegt auf den Schultern der Anbieter. Sie müssen Produkte anbieten, die reale Bedürfnisse befriedigen, nicht nur ihre eigenen und nicht nur die ihrer Angestellten. Sie müssen ihre Geschäftsaktivitäten nach den Bedürfnissen des Marktes ausrichten, den sie bedienen wollen. Zur Zeit ist es offensichtlich, daß sie ihre Geschäftsziele an einer kleinen Anzahl von Kunden ausrichten mit beschränkten Bedürfnissen. Für mich ist das ein Trauerspiel, weil die Forth Hersteller freiwillig, so wie sie ihre Geschäfte führen, ihre Dienstleistungsrolle so engstirnig definieren.

Vielleicht entstehen einige der Innovationen, die du vorhin erwähnt hast, dadurch, daß die Leute erkennen, was gebraucht wird ...

Genau. Als du dies gesagt hast und du warst an der Stelle "Innovationen", da bin ich zusammengefahren - und als du bei "was gebraucht wird" warst, da habe ich ein bißchen gelächelt, weil es genau das ist. Ich glaube, daß wir bisher einen wertvollen Edelstein geputzt haben und aller Welt erzählt haben, wie toll er ist, statt uns darum zu kümmern, welche Art Schmuck die Leute eigentlich haben wollen. Ich will keineswegs propagieren, daß wir die Grundsätze von Simplizität, Portabilität und Allgemeingültigkeit über Bord werfen, die die Stärke von Forth ausmachen. Aber wir

haben eine ungeheuer wertvolle Grundlage, die wir bisher nicht verwendet haben, um auf die Anforderungen der Benutzer einzugehen. Einige Anbieter haben in der Vergangenheit versucht, in dieser Richtung aktiv zu werden, aber es scheint mir, daß die Wachstumsrate der Computer Industrie, die zwischen 35 - 50 Prozent liegt, nicht auf die Anbieter in der Forth Community durchgeschlagen ist.

Kann jemand, der Forth Anbieter werden will, mit verlockenden Gewinnen rechnen, oder ist das voreilige Wunschdenken?

Ich sehe eine Möglichkeit groß wie ein Heuschäber, ein riesiger Haufen Gold am Ende des Regenbogens oder Ali Babas Schatzhöhle. Es ist die ungeheuer große finanzielle Möglichkeit, die in Forth steckt, und die unglücklicherweise nicht ausgenutzt wird. Die meisten scheinen drei Jahre alte Probleme mit zwei Jahre alten Methoden zu lösen. Sie kümmern sich nicht darum was heutzutage passiert und was gebraucht wird.

Ich werde einige Hinweise geben, was meiner Ansicht nach Bestandteile davon sind. Ich stelle mir ein System vor, das eine integrierte Datei- und Datenbankstruktur enthält, die nach zeitgemäßen Entwurfsrichtlinien entwickelt wurde so, wie man sie in Universitätsveröffentlichungen nachlesen kann. Komponenten hiervon wären zweiphasiges Recordlocking und Transaktionen mit vorwärts/rückwärts Eintragungen. Das muß dann mit einer kompatiblen Querysprache und virtueller Execution verbunden werden. Nicht der forthartige Zugriff auf Massenspeicherblöcke und Datenspeicher, sondern die Möglichkeit, Code aus dem virtuellen Speicher heraus zu auszuführen. Eine ausgewogene Kombination aus diesen Elementen wäre auf dem Markt der Betriebssysteme wettbewerbsfähig im Vergleich zu UNIX zum Beispiel.

Ich bemerke Anbieter wie Unisoft und Relational Technology in Berkeley, die sehr qualifizierte Arbeit bieten und heute vier bis fünf Millionen Umsatz erzielen und auf die fünfzehn bis zwanzig Millionen zusteuern. Wenn der Code einmal geschrieben ist, gibt es eine Vielzahl von Verwendungsmöglichkeiten, und man kann Lizenzen an einige hundert andere Firmen verkaufen, dieses System zu vertreiben. Diese zwei Firmen erreichen aus dem Stand zwei bzw. zwanzig Millionen Umsatz pro Jahr, und ich sehe keinen Grund,

warum das nicht auch mit Forth als Grundlage erreicht werden könnte. Kurz, nach diesen grandiosen Bemerkungen, wie kann es erreicht werden, daß die Anbieter ihre Ziele höher setzen, daß sie ihre Wünsche hoch genug schrauben, um etwas Ähnliches zu erreichen?

Es ist eine große Herausforderung. Die technologischen Grundlagen sind vorhanden - es muß also scheinbar nur noch der Zufall zu Hilfe kommen. Vielleicht brauchen wir nur etwas Glück, daß ein kompetenter Erfinder mit einem sicheren Gespür für Kundenwünsche in eine passende Geschäftsumgebung gerät, um so etwas auszulösen.

Dennoch scheuen viele, die Forth benutzen, große Systeme

.....
Ich spreche nicht von großen Systemen. Das Betriebssystem Pick hat viele der Eigenschaften, die ich eben genannt hatte, und ist kompakter als Forth. Es läuft mit einem Kern von 4 K und 250 K Code. Es hat virtuelle Ausführung, eine Datenbank mit Querysprache, ein Textverarbeitungssystem und Multitasking. Für jemanden, der das System benutzen will, bedeutet es eine Millionen Dollar teure Lizenz. Das ist Führertum. Ihre Kunden stehen Schlange. Früher schafften sie eine Implementation pro Jahr, heute sind es vier und die Kunden stehen Schlange und warten. Wenn Dick Pick und Chuck Moore am Anfang zusammengekommen wären, dann hätte ein Wunder geschehen können.

Warum passiert so etwas nicht mit Forth? Ich weiß es auch nicht genau. Vielleicht deshalb, weil Erfindungen dann gemacht werden, wenn Leute sich in unbefriedigenden Situationen befinden, wenn sie sich in die Ecke gedrängt fühlen. Dann müssen Lösungen gefunden werden - seien diese nun technischer, organisatorischer, menschlicher oder finanzieller Art. Vielleicht müssen wir einfach nur so weitermachen wie bisher, bis sich die richtige Mischung von Leuten in einer solchen Ecke befindet.

Wenn ich hier so klug rede, warum mach ich es dann nicht selber? Hm, es ist eben nicht einfach; es ist die herkulische Aufgabe, eine Industrie aufzubauen. Bisher haben Forth und Chuck Moore nur eine Nebenbeschäftigung erzeugt. Wie gelingt es uns, daraus eine Industrie zu machen? Das ist der Grund, warum ich auf der letzten FORML Konferenz den Beitrag über Industrieführerschaft gegeben habe; ich wollte damit inspirieren.

Forth ist heute wie ein Paar gemütlicher Pantoffeln - angenehm, einfach, Spaß und so weiter. Es muß sich eine neue Ordnung herausbilden, um auf Anforderungen zu reagieren. Wir brauchen eine neue Eiszeit, ein paar mehr frierende Leute, die darum kämpfen, einen Mammut fürs Abendessen zu erlegen.

Was sind deine eigenen Erwartungen an FIG?

Ich bin hier seit zu langer Zeit der Steuermann. Das hilft weder FIG weiter, noch mir selber. Deshalb müssen Wege gefunden werden, mich zu ersetzen. Dies scheint mit am besten dadurch möglich, daß entwicklungsfähige Menschen sich einstellen, die auf eh-

renamtlicher Basis teilnehmen wollen und dabei ihre Fähigkeiten entwickeln. Ich glaube, daß das zur Zeit vonnöten ist. Es ist wie beim Staffellauf und wir brauchen jetzt die nächste Staffel. Daß John Hall sich eingefunden hat, ist schon die erste frische Brise. Er kam einfach zum richtigen Zeitpunkt, hat die Herausforderung angenommen und hält sich recht wacker. Vor zwei oder drei Jahren hat Kim Harris so etwas mit FORML vorexerziert. Jetzt bräuchten wir eigentlich etwas Verstärkung für organisatorische Routinearbeiten. Für einige Zeit hatte ich das übernommen, dann machte es Roy Martens und jetzt scheinen wir die nächste Generation auf dem Gebiet zu brauchen.

Für alles andere, was darüber hinaus geht, fallen mir nur Platinen ein. Wir wollen eine Mitgliederorganisation mit genügend internationaler Verbreitung, um uns alle gegenseitig zu unterstützen. Also, wir brauchen die notwendigen menschlichen, technischen und kommunikativen Fähigkeiten, um Forth Dimensions am Blühen zu erhalten und die Chapter miteinander auf persönlicher und offizieller Ebene kommunizieren zu lassen. Dann brauchen wir jährlich ein bis zwei internationale Veranstaltungen, um auf einer globalen Ebene zu interagieren. Es ist wie bei Forth Worten: Es gibt die Codeworte und die High-level Worte und die Interaktion mit dem Benutzer ganz oben, alles in einzelnen Schichten. Genauso ist es mit den Gruppenaktivitäten auf lokalem, nationalen und internationalen Gebiet. Die Forth Interest Group hat einen solchen Grad von Unterstützung und Anforderungen erreicht, daß sie unvermindert weitermachen wird und sich aus ihrer Eigenenergie heraus weiterentwickeln wird.

Gibt es Grenzen des Wachstums? Ich bin mir da nicht sicher. Es muß wohl so sein - wir werden niemals eine Million Mitglieder haben. Ich benutze das Buch Starting Forth, um unser mögliches Universum abzugrenzen: 50.000 bis 60.000 Exemplare sind gedruckt und jeder, der es besitzt, sollte auch FIG-Mitglied sein. Wenn wir das erreicht haben, sind wir an unserer Grenze. Mit einer solch großen Gruppe posich die Möglichkei-

--- BEZUGSQUELLEN FÜR FORTHINSTALLATIONEN ---

Die Mehrzahl der Reaktionen auf die Vorstellung der FORTH-Gesellschaft im C^T lauteten in etwa so:

"Wo bekomme ich ein FORTH-System für meinen Computer". Woraufhin wir erstmal alle uns bekannten Computerzeitschriften nach Angeboten durchsuchten, die nicht älter als ein halbes Jahr sind. Diese Angebote sind natürlich noch unverbindlich. Brauchbare Aussagen über die Vor- und Nachteile der Systeme im einzelnen können wir erst nach der Auswertung unserer Fragebogenaktion, die sich auf bereits implementierte FORTH-Systeme bezieht, machen. Also für alle die's eilig haben, hier nun ein paar Adressen (zum Testen sozusagen).

- Für SHARP PC 1500 ---- PC 1500-FORTH (CE 155 erforderl.)
Cassette + Handbuch 129.- DM
-- A.Ulrich --
Kopernikusstraße 11
8000 München 80
- Für C 64 ---- C 64-FORTH auf Cassette 49.- DM
Handbuch 25.- DM
-- D.Luda --
Staudingerstraße 65
8000 München 83
- FORTH 64-Modul 198.- DM
-- bst-Computronic --
Burgstraße 126 a
6000 Frankfurt 60
tel.:0611/45 38 57
- Für VC 20 ---- VC 20 FORTH-Modul mit Screeneditor,
Disk, Utilities
-- FORTH-QUELLE Angelika Flesch --
Schützenstraße 3
7820 Titisee-Neustadt
- Für APPLE u.komp. ---- FORTH mit Assembler und Editor sowie
Grafikpaket auf Cassette 80.- DM
-- J.Kaiser --
Hausmattenweg 5
7801 Oberried
- Für TI 99/4A ---- FORTH (verbesserte F.I.G. 1.0 Vers.)
auf Disk/Cass. mit Handbuch 120.-
Utilities 50.-, EPROMS für Modulplat.
170.-, Source-Listing dazu 50.-
Modulplat.leer 30.-
-- Dipl.Ing. M.Weiland --
Feldgärtenstraße 50
5000 Köln 60
- SOURCE LISTINGS ---- Assemblerlistings 35.- DM
(für alle gängigen Install.Manual 35.- DM
Prozessoren) -- C.Schmidt Software --
Bungestraße 8
3500 Kassel

Was sind das für Leute, die sich von Forth angezogen fühlen, und welche Art von Leute hat damit Schwierigkeiten?

Ich sehe zwei verschiedene Typen, die Forth als wertvoll empfinden. Erstens die erfindertischen Menschen, ein Typus von Forschern, die genauso viel Genuß aus der Art der Lösung eines Problems erzielen wie aus der Lösung des Problems selber. Ich kann ein Programm in BASIC, Fortran oder Forth schreiben; am stolzesten bin ich immer über die Lösung in Forth. Das ist die, die ich anderen zeigen würde. Der andere Typus sind diejenigen, denen die Flexibilität von Forth gefällt. Manchmal sind die ein bißchen rebellischer und nonkonformistischer als andere, sie wollen ein Problem nicht so wie ein anderer lösen. Sie benutzen die Freiheit, die Forth bietet, um ihren eigenen Lösungsstil und ihre eigene Ausdrucksweise zu entwickeln. Statt ihre Lösung rumzuzeigen, sind sie damit zufrieden, daß sie die Aufgabe auf ihre Art gelöst haben. Die letzteren ist auch etwas häretischer bzw. nonkonformistischer in der Beziehung, daß sie am stolzesten über die Lösung sind, wenn sie etwas anders als üblich ist.

Was sind deine Vorurteile als Forth Programmierer?

Ich habe das gleiche un-gute Vorurteil, wie die meisten: Daß ich es besser kann, daß mein Code von vornherein besser als dein Code ist. Ich habe genug Abstand, um darüber zu witzeln und es zu bemerken. Aber es kommt andauernd vor, daß das Rad neu erfunden wird, und man glaubt, daß die einzig wahre Lösung die ist, die einem gerade eingefallen ist. Forth unterstützt eine solche Denkweise. Es bietet die Grundlage und erschließt uns die Einsichten kluger und kreativer Leute. Andererseits bedeutet es, daß wir die Fähigkeit verlieren, auf der Arbeit anderer aufzubauen. Ich bemerke die verbreite Tendenz, daß jeder sein eigenes System schreiben oder zumindest installieren will, statt bei den Anbietern nach einem geeigneten System zu suchen. Wenn mehr Leute die Angebote der Anbieter wahrnehmen, dann gäbe es bessere Systeme und einen besseren Markt.

Ein Beispiel dafür: Wenn ich in eine Zeitschrift wie Dr. Dobbs's Journal schaue, dann finde ich da sechs Angebote für C Compiler; einen für 350,- Dollar und die anderen für zwischen 500,- und 600,- Dollar. Danach suche ich in

derselben Zeitschrift nach Forth Systemen und das billigste kostet 35,- und das teuerste, glaub ich, 195,- Dollar, die meisten zwischen 85,- und 100,- Dollar. Also, ein Preis wird oft als Qualitätsmerkmal genommen, und daraus erfährt ein möglicher Kunde, daß ein Forth System ihm höchstens ein Fünftel der Möglichkeiten eines C Compilers bietet. Das ist keine gute Aussage. Wir müssen den Wert von Forth Systemen deutlich machen, so daß sie zumindest genauso wertvoll wie C Systeme erscheinen.

Vorurteil 2: "Ich kann es besser", ist typisch für die Individualisteneinstellung in FIG. Wir hatten neulich ein Planungstreffen auf dem diese Aspekte behandelt wurden, und es wurde deutlich, daß eines der Charakteristiken eines Forth Programmierers darin besteht, daß er lieber versucht, die Programmierungsumgebung seinen Bedürfnissen anzupassen, als seine Bedürfnisse herabzuschrauben und sich anzupassen. Ich glaube nicht, daß das unbedingt schlecht ist, aber wir sollten uns dessen bewußt sein. Wo können wir dies als unsere Stärke einsetzen, um darauf aufzubauen?

Es heißt, daß im Zusammenhang mit Programmierung besser nicht unbedingt am klügsten ist. Verfeinerung des Entwurfs und des Codes können sich endlos hinziehen, so daß das Projekt nie zum Abschluß kommt.

Dafür gibts einen klassischen Spruch: Hardware Entwickler und Ingenieure sind daran gewöhnt, sich gegenseitig auf den Schultern zu stehen; Komponenten zu benutzen, die von anderen gefertigt werden. Programmierer stehen sich gewöhnlich auf den Zehen. Sie kommen sich gegenseitig ins Gehege und jeder einzelne bringt das Gesamtprojekt nur geringfügig weiter. Wie kommen wir dahin, daß wir einander auf den Schultern stehen, so daß jeder dadurch eine Verdoppelung seiner Produktivität erfährt?

Die Menschen in der Forth Interest Group haben sich geändert und die kollektive Struktur der Gruppe hat sich geändert. In der Ära von 1979-1980 waren wir sehr stark in einer missionarischen Rolle. Wir trafen uns und sagten: "Schau dir das hier an, lies meinen Code, Forth ist einfach toll". Es war aus Stolz heraus, und es ist erklärbar aus der Tatsache, daß die meisten vorher noch nie mit Forth konfrontiert waren. Ich bin sehr zufrieden, daß sich das geändert hat, und daß wir heute in unserem Ton nüchterner sind. Heute versorgen wir eine aufmerksame Zuhörerschaft mit Information, die an uns herantritt mit der Frage: "Hör mal, ich hab von Forth gehört. Könnt ihr mir helfen, es zu verstehen und es für mich zu nutzen?" Das ist ein bemerkenswerter Fortschritt in der kurzen Spanne von drei Jahren.

Übersetzt
aus FORTH DIMENSION V/6
von K.Schleisiek

DAS RÄTSEL DES MONATS

Gesucht ist die Definition des neuen Forth - Wortes erst . Erst hinterläßt eine Zahl auf dem Stack. Vor der Ausführung weiß erst bereits, was für ein Ergebnis produziert werden wird und es benutzt diese prognostische Vermutung, um die Zahl zu berechnen.

: erst (- Zahl) ??? ;

Dem Gewinner winkt eine bahnbrechende Erfindung. Lösungen nimmt die Redaktion entgegen.

DO-LOOP



DIE FORTH-QUELLE ANGELIKA FLESCH

Mit Erscheinen dieser ersten Nummer der "Vierte Dimension" möchten auch wir es nicht versäumen, uns bei Ihnen vorzustellen.

Unter anderem bieten wir FORTH-Systeme der Firmen
Laboratory Microsystems Inc. (LMI)
Micro Motion (MM)
Mountain View Press Inc. (MVP), alle USA, an.

Außerdem führen wir für alle gängigen Rechner- und Prozessortypen fertige FORTH-Systeme sowie Assemblerlistings. Des Weiteren erhalten Sie bei uns ein komplettes, ständig aktualisiertes Angebot an FORTH Literatur.

Aufgrund dieser Tatsachen sind wir der derzeit größte Anbieter von FORTH Artikeln im deutschsprachigen Raum.

Ein kleiner Auszug aus unserem Lieferprogramm:

C64 FORTH Diskette mit Grafik ca. 100K Source	99,--	DM
C64 16K-Modul mit Sprites u. Sound etc.	198,--	DM
VC20 8K-Modul mit Editor, Kass. u. Disk supp.	188,10	DM
CBM 3032/4032/8032, 4040/8050 Format Dt. Handb.	248,--	DM
Spectrum FORTH, Kassettensupport	79,--	DM
HX20 FORTH (Eproms) mit Handbuch	188,10	DM
APPLE II MasterFORTH v. MM	376,20	DM
Die gängigsten CP/M Systeme (8080,280,8086,68000)	376,20	DM
FIG-Listing für 1802,6502,6800,8080,8086,9900,6809,280, 68000,28000,APPLE,NOVA,PDP11,VAX,ECLIPSE.	je 48,--	DM
Installationmanual, prozessorunabhängig	48,--	DM
FORTH-83 Listings für 6502,8080,8086	je 88,20	DM
Starting FORTH von Leo Brodie	55,60	DM
Deutsche Übersetzung in Kürze lieferbar: Programmieren in FORTH von Leo Brodie	48,--	DM
System Guide to FIG-FORTH von Ting	88,20	DM
Pocket Guide to FORTH	25,--	DM

Natürlich leisten wir auch qualifizierte Hilfestellung bei der Installation aller bei uns gekauften Systeme, sowie bei allen eventuell auftretenden Fragen und Problemen.

Fordern Sie bei Interesse einfach unseren kompletten Katalog sowie unsere neueste Preisliste an, die wir Ihnen gerne gratis zusenden.

Die hier angegebenen Preise verstehen sich bei Redaktionsschluß.

DIE FORTH-QUELLE ANGELIKA FLESCH
Schützenstr. 3, 7820 Titisee-Neustadt
Tel. 07651/1665

Fragebogen 1/84

Mit diesem Fragebogen wollen wir ein Forum einrichten , um die unterschiedlichen (??), erhältlichen FORTH - Versionen anhand praktischer Erfahrungen beurteilen zu können. Solange der '83 er Standard noch nicht fertig ist, hoffen wir Euch mit der Auswertung der Antworten wenigsten vor allzu schlechten/teuren oder sonstwie ungeigneten Implementationen bewahren zu können. Versucht also bitte objektiv zu sein, damit wir hier nichts falsches in Umlauf setzten.

Meine Erfahrungen mit FORTH-IMPLEMENTATIONEN würde ich wie folgt bewerten:

Anbieter	Version	Jahr	Preis
1. _____	_____	_____	_____

Prozessor: _____

Doku: gut/mittlm./ungenügend/keine/unleserlich

zusätzlich über FIG-FORTH hinaus:

<input type="checkbox"/> Floating Point	<input type="checkbox"/> Trace
<input type="checkbox"/> Graphik	<input type="checkbox"/> Fullscreen Editor
<input type="checkbox"/> String Stack	<input type="checkbox"/> Dateiverwaltung
<input type="checkbox"/> OS-Interface	<input type="checkbox"/> Utilities/welche _____

Anbieter	Version	Jahr	Preis
_____	_____	_____	_____

Prozessor: _____

Doku: gut/mittelm./ungenügend/keine/unleserlich

zusätzlich über FIG-FORTH hinaus:

<input type="checkbox"/> Floating Point	<input type="checkbox"/> Trace
<input type="checkbox"/> Graphik	<input type="checkbox"/> Fullscreen Editor
<input type="checkbox"/> String Stack	<input type="checkbox"/> Dateiverwaltung
<input type="checkbox"/> OS-Interface	<input type="checkbox"/> Utilities/welche _____

Name :
Vorname:
Strasse:
PLZ/Ort:
Tel. :

Sonstiges: _____

Common Interface Alpha
Holstenstraße 191
2000 Hamburg 50 tel. 040/ 43 50 70

EINSTIEG

Um auch weiterhin die **VIERTE DIMENSION** zu erhalten und an dem Informationsnetzwerk teilzuhaben übersende ich ihnen den Betrag

von: 23,- DM (Einzel) 55,- DM (Firma) _____,- DM (Sponsor)

Ich erhalte dafür die nächsten 4 Ausgaben der "**VIERTE DIMENSION**".

UNTERSTÜTZUNG

() - Ich unterstütze die **FORTH GESELLSCHAFT** mit einer **SPENDE** von _____,- DM .

() - Ich möchte die **FORTH GESELLSCHAFT** regional unterstützen, indem ich mich als Ansprechpartner für die **REGION** _____ (PLZ-Gebiete) zur Verfügung stelle.

() - Ich werde die **FORTH GESELLSCHAFT** unterstützen indem ich Artikel oder Buchbesprechungen für die **VIERTE DIMENSION** schreibe resp. ihnen meine bisher geschriebenen Artikel zusende. Das Copyright für diese Artikel liegt bei mir.

() - Ich möchte die **FORTH GESELLSCHAFT** unterstützen, und mich als Ansprechpartner bzw Spezialist für folgende Computer / Prozessoren / Betriebssysteme zur Verfügung stelle :

() - Ich habe ausserdem noch folgende Erfahrungen, Kenntnisse die für die Verbreitung von **FORTH** von Interesse sein könnten :

Den Betrag von _____,- DM habe ich am _____ auf euer
Konto überwiesen: ()
als Verrechnungsscheck beigelegt : ()
in Briefmarken beigelegt: ()

Name : _____

Strasse : _____

PLZ/Ort : _____

Tel. : _____/_____

BITTE ABTRENKEN UND EINSENDEN AN DIE FORTH - GESELLSCHAFT