

VIERTE DIMENSION

3/1992

8. Jahrgang 1992 3. Quartal DM 7,50

FRP 1600

FORTH RISC-PROZESSOR

Forth out?
Forth in!

Fourieranalyse

Forth und der Rest der Welt
Goppold-Kolumne

F-PC mit Ö's

ünd ändern SönderBeichen

Happy Mailboxing

**Linked Actions
und kein Ende**

Sägewerk
gesteuert von Forth

WANTED
in
Forth!

F68K
das erste Jahr

LPT1 SWAP
LPT2

FORTH

MAGAZIN

Organ der FORTH-Gesellschaft e.V.

SMAN - Der Software-Manager

Probleme mit der Verwaltung großer Mengen Quelltext?
Rasches Finden von Quelltext-Modulen nicht möglich?
Zusammenfügen von Modulen umständlich?
Keine einheitliche Umgebung für verschiedene Compiler?

SMAN kann's !

DFE-Team, Frank Stüss
An der Turnhalle 6
6369 Schöneck 2
Tel.: 06187-91503
FAX: 06187-91725

F-PC-ak version 4.0

Der wesentliche Sprung nach vorne.

DAS Forth für den PC.

Ideal für Forth- und/oder F-PC-Anfänger,
Assembler-Anfänger,
sowie Online Steuerungen, Unterricht.

2. Platz im EchtZeit'92 ProgrammierWettbewerb.

Komfortable Bedienung, per Maus und Menü,
umschaltbar ähnlich Borland/WordStar/NC.
Keine Probleme mehr mit Tastatur & Zeichensatz
(MF2-Deutsch, Umlaute, griechisch ..).
Hilfe in Bedienung/Umfang/Fehlern verbessert.
Viele bugs and quirks und Hilfe-'Löcher' behoben.
Codedebugging nun auch in kritischen Fällen.
TCOM und externe Processoren im HilfeZugriff.
Als universeller HypertextGenerator nutzbar
(BedienungsAnleitungen usw.).

Zusätzlich fünf Floppies 1M44:
F-PC-'INSTANT'-Floppy, F-PC Original v.3.5610,
TCOM v.2.17/2.26, andere Forth-Systeme (PC und Controller),
ausgesuchte DOS-Utilities (inkl. MODEM-Prog. für FORTH-Mailbox)

3-bändige Dokumentation lieferbar.

Aktionspreise für Updates bis Ende 1992.

Arndt Klingelberg \ StrassburgerStr. 12 \ D-W 5110 Alsdorf
Tel. 0+2404 - 6 16 48 Fax. 0+2404 - 6 30 39

Info anfordern !
Auf IBM-PC, Amiga, Atari . . .

embedded
FORTH für den
6502

Einplatinencomputer schneller
programmieren mit F65-FORTH

R65C02 65SC02 W65C02



Rafael Deliano
Steinbergstr. 37
8034 Germering
089 / 84 18 317



IMPRESSUM

Name der Zeitschrift

VIERTE DIMENSION
FORTH MAGAZIN
Organ der Forth-Gesellschaft e.V.

Herausgeber

Forth-Gesellschaft e.V.
W-8044 Unterschleißheim

Redaktionsleitung

Rolf Kretzschmar (rk), (verantwortlich)
Rote Gasse 7, W-5112 Baesweiler
(Redaktionsadresse)
Tel/Fax: 02401-88891

Redaktion

Arndt Klingelberg (akg), Alsdorf
Tel.: 02404-61648 Fax: 02404-63039
Klaus-Peter Schleisiek (kps), Aachen
Tel/Fax: 0241-873462

Layout, Satz, Herstellung

ORGA Sport, Rilkestr. 8, W-5110 Alsdorf
Tel/Fax: 02404-61425

Grafik, Illustration, Layout

Rolf Kretzschmar (rolf)

Anzeigenverwaltung

Forth-Büro, Postfach 1110
W-8044 Unterschleißheim
Tel.: 089-3173784 oder
Forth-Mailbox Tel. 089-8714548 8N1

Redaktionsschluß

Feb., Mai, Aug., Nov.

Erscheinungsweise

vierteljährlich

Auflage

1000

Preis

Einzelheft DM 7,50, Abonnementpreis
DM 40,-, bei Auslandsadresse DM 45,-
inklusive Versandkosten

Manuskripte und Rechte

Berücksichtigt werden alle eingesandten Manuskripte von Mitgliedern und Nichtmitgliedern. Leserbriefe können ohne Rücksprache gekürzt wiedergegeben werden. Beiträge der Redaktion sind vom jeweiligen Redakteur mit seinem Kürzel (s.o.) gekennzeichnet. 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 auszugsweise 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, Aufbauskiizen 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.



Die Wende

von Rolf Kretzschmar

Der Stapel an Beiträgen wächst stetig. Die meisten Autoren haben an die Einleitung und das Stichwörterverzeichnis gedacht. Das Material reicht endlich, um auch Themenhefte zu realisieren. Viele Autoren geben ein Paßfoto mit persönlichen Daten bei. Die Länge der Artikel ist in der Regel für ein 40-Seiten-Heft angemessen. Von zehn potentiellen Autoren, die ich anschreibe oder anrufe, schicken acht einen Beitrag. Die Teilnehmer der Forth-Tagung sind besonders aktiv. Spontan sind neue regionale Gruppen gegründet worden, so daß die eine Seite in der VD nicht ausreicht, alle Adressen aufzuzählen. Außer Moers möchten nun drei weitere Gruppen mit Berichten ihrer Aktivitäten unser Heft anreichern. Die Kontakte zur mc zahlen sich aus: Die Redaktion hat auf meinen Brief umgehend geantwortet und wird eine ganzseitige Anzeige schalten. Darüber hinaus wird uns gestattet, Quellcode als Paperdisk (lesbar mit einem Scanner) abzdrukken. Die Verhandlungen mit einer Vertriebsgesellschaft, die Zeitschriften auch gezielt an Bahnhofs- und Flughafenkioske ausliefert, sind in ein erfreuliches Stadium gelangt. Die erste Lehrer-Fort(h)-bildungsveranstaltung beginnt im nächsten Schuljahr. Forth soll für den Einsatz an Berufsschulen gefördert und empfohlen werden. Die Mitgliederzahl der Forth-Gesellschaft hat die 1000-er Marke überschritten...

... und dann wachte ich auf und mußte lächeln. Hatte doch mein Traumgenerator einen erkennbaren Aufwärtstrend schamlos aufgeblasen. Kommentar von Jörg Staben: "Wart's ab, Du wirst es erleben..."

Euer

Mitteilung aus dem Forth-Büro

von Ulrike Schnitter

VD-Versand

Im Leserbrief von Friederich Prinz, Moers, **VIERTE DIMENSION** 8/2, wurde kritisiert, daß die VD 8/2 zu sehr unterschiedlichen Zeiten bei den Mitgliedern eingetroffen sei.

Wir möchten deshalb unseren Mitgliedern erläutern, wie eine Ausgabe der VD zum Versand gebracht wird und wo es unter Umständen zu Verzögerungen kommen kann. Wir wählen die für die Gesellschaft günstigste Versandform, nämlich **DRUCKSACHE**.

Sobald das **FORTH-Büro** von der Druckerei erfährt, daß die Ausgabe in den nächsten Tagen abgeholt werden kann, wird ein Datenbankauszug, nach Postleitzahlen sortiert, erstellt und eine Versandliste ausgedruckt. Nach Überprüfung der Versandliste werden mit dem gleichen Datensatz die Etiketten ausgedruckt. Nun werden die C4-Briefumschläge frankiert und mit den Etiketten versehen. Die VD wird termingerecht abgeholt (seit Heft 8/2 per Bahn-Stückgut zugestellt), die entsprechenden Beilagen einsortiert und in die Briefumschläge eingelegt. Grundsätzlich wird die VD nach nur einem Tag Verweildauer im **FORTH-Büro** insgesamt zum Versand gebracht. Ausnahmen gibt es beim Heft 4 eines Jahrgangs, wenn die Beilagen für das Heft zusätzlich bearbeitet werden müssen. So wird in Heft 4 in der Regel die aktuelle Mitgliederliste (nur für Mitglieder), die Einladung zur **FORTH-Tagung** und ein vom **FORTH-Büro** teilweise individuell ausgefüllter Überweisungsauftrag des Postgiroamtes Hamburg einsortiert. Diese Arbeit nimmt trotz Überstunden bis spät in die Nacht so viel Zeit in Anspruch, daß diese Ausgabe an zwei aufeinander folgenden Werktagen zur Post gebracht wird. Bei diesem Heft sind fördernde Mitglieder und die Mitglieder, die uns eine Ermächtigung zum Einzug des Mitgliedsbeitrags per Lastschrift er-

teilt haben, im Vorteil. Diese Hefte kommen zuerst zum Versand, da hier kein Überweisungsauftrag ausgefüllt und einsortiert werden muß. Fördernde Mitglieder erhalten im Januar eine Rechnung über ihren Mitgliedsbeitrag.

Noch einige grundsätzliche Bemerkungen zum Versand als **DRUCKSACHE**: Wenn wir 400 oder mehr Drucksachen auf einmal zum Postamt bringen, werden diese häufig an ein spezielles Postamt in München zum Stempeln und/oder Sortieren weitergeleitet. Um dies zu vermeiden, haben wir den Versand auch schon geteilt und in zwei verschiedenen Postämtern eingeliefert. **DRUCKSACHE** wird bei der Post auch nicht bevorzugt behandelt, so daß sich durchaus Laufzeitunterschiede von bis zu einer Woche ergeben können. (Zwei Tage, trotz Streik bei der Deutschen Bundespost, wie im Leserbrief angegeben, bewegen sich innerhalb der normalen Postlaufzeiten).

Allerdings klappt es mit der Bundes-



post nicht immer so, wie wir uns das wünschen. So wurden wir schon ein paarmal von Mitgliedern (vor allem aus den neuen Bundesländern) 3-4 Wochen nach dem Versand angerufen, wo denn die VD geblieben sei. Sofortiges Nachsenden der VD hatte dann zur Folge, daß das Mitglied zwei VDs zur gleichen Zeit erhielt. Zur Ehrenrettung der Deutschen Bundespost möchten wir darauf hinweisen, daß nach unserem Wissen in der Zeit von 1988 bis heute bei ca. 8000 Aussendungen nur zwei (!) VDs auf dem Versandweg verlorengegangen sind.

□

Das Direktorium berichtet

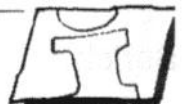
Das Direktorium hat sich getroffen, jede Menge Milch getrunken und überlegt. Was hat sich so getan? Was tut sich gerade? Was soll in nächster Zeit geschehen?

Zunächst einmal gibt's da noch ein paar mögliche Mißverständnisse auszuräumen. Was soll denn das mit dem Forth-Stempel (vgl. VD 8/2 S.25)?! Die zu stempelnden Forth-Systeme sollten bestimmten Anforderungen genügen. Wie bitte? Ja genau! Hierbei handelt es sich aber nicht um eine Bewertung der Qualität, sondern um die Tatsache, daß der Autor - ein Mitglied der Forth-Gesellschaft - den Sourcecode, die Doku und das Vertriebsrecht der Forth Gesellschaft zur Verfügung stellt. Beispiele dafür sind das volks**FORTH** (C64, CP/M, PC, Atari), das **FG-Forth** (RTX-

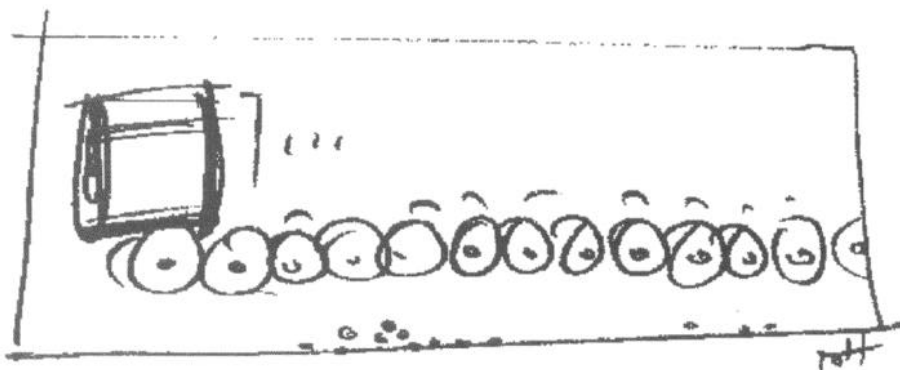
2000) und das **F68K** (portabel über alle 68000er). Ein solcher Beitrag, um die finanzielle Situation der Forth Gesellschaft zu verbessern, muß nicht unbedingt ein Forth-System sein; die **FG** nimmt selbstverständlich auch gerne Bücher, Dienstprogramme und anderes in ihren Vertrieb auf.

Erfreulich ist, daß neben zunehmender Akzeptanz von Forth als selbstverständlichem Werkzeug Forth immer mehr als Ausbildungssprache eingesetzt wird. Das zeigt sich darin, daß gerade Schulen sich für Forth und für die **FG** interessieren. Über die Aktivitäten, die mehr denn je von der Zusammenarbeit der Mitglieder der **FG** getragen werden, wird in der VD an jeweiliger Stelle berichtet.

□



Impressum	1
Editorial Die Wende	<i>Rolf Kretzschmar</i>	1
Mitteilung aus dem Forth-Büro VD-Versand	<i>Ulrike Schnitter</i>	2
Das Direktorium berichtet	2
Inhaltsverzeichnis	3
Automation eines Sägewerkes Ein Erlebnisbericht	<i>Ralf Neuthe</i>	4
Internationaler Zeichensatz im F-PC F-PC wird auf 8 Bit aufgebohrt	<i>Klaus Vogt</i>	7
Fourieranalyse in Forth	<i>Klaus Kohl</i>	13
LINKED ACTIONS und kein Ende	<i>Jörg Staben</i>	16
F68K - das erste Jahr	<i>Jörg Plewe</i>	17
Ein klein, fein Helferlein Den Screen printen aus F-PC	<i>Jörg Staben</i>	22
F68K - Neuigkeiten Ergänzungen zu F68K - das erste Jahr	<i>Jörg Plewe</i>	22
FORTH RISC-PROZESSOR FRP 1600 Produktbesprechung	<i>Arndt Klingelberg</i>	23
Forth und der Rest der Welt Kolumne	<i>Andreas Goppold</i>	25
Swopp mal wieder LPT1 und LPT2 vertauschen	<i>Jörg Staben</i>	27
Buchbesprechung Theorie der Betriebssysteme - Eine Einführung in die Koordination paralleler Prozesse Prof. Dr. mult. Dieter Werner	<i>Heinz Schnitter</i>	29
Wanted in Forth! Das was es woanders schon gibt, wollen wir auch in Forth.	<i>Jörg Staben</i>	30
Leserbriefe	32
Forth in Mailboxen	35
Gruppen, Fachberatung, Ansprechpartner	36



Automation eines Sägewerks mit Forth

Ein Erlebnisbericht von Ralf Neuthe

Neben der staubigen Umgebung gibt es noch andere Probleme, die einen Forth-Programmierer erwarten, wenn er den Steuercomputer eines Sägewerks anpassen soll. Sie erfahren in dem Bericht, welchen Einfluß zum Beispiel "kurzes Holz" auf die Psyche des Programmierers haben kann.

Zur Vorgeschichte

Alles fängt einmal an, und so begann diese Geschichte mit einem Telefonanruf im Januar 1991 bei der Firma FORTECH Software GmbH, die mir seinerzeit Lohn und Brot gab.

Am anderen Ende der Leitung war eine Firma, die Sägewerke umrüstet, um den Holzdurchsatz zu steigern. Das frisch gebackene Unternehmen hatte zum ersten Mal im größeren Umfang die Technik seines österreichischen Partners zu installieren. Diese erstreckte sich von der Vermessung der Baumstämme bis zur für jedes Sägewerk spezifischen Steuerung mittels eines Forth-Rechners.

Die Uhren standen dort schon fast auf 5 vor 12 und es war immer noch kein Programmierer gefunden, der in der Lage war, diesen Rechner zu programmieren. Das war die große Stunde von FORTECH (Ausführende: Dr. Woitzel und ich).

Das Arbeitsfeld

Es existierte ein speziell für diese Aufgabenstellung zusammengestellter VME-Bus-Rechner, der für die Simulation des Prozesses mit einer Sammlung von Lampen und Schaltern ausgerüstet war. Nach dem Einschalten bootete ein 68000er 32bitiges Forth-System, das fast dem 83er Standard gehorchte. Es war mit einem Round-Robbing Multitasker ausgestattet. Das ganze Forthsystem hatte zwar seine Eigenheiten, es lief aber hervorragend stabil.

Der Rechner sollte mit seinen Ein- und Ausgängen einen Rundholzplatz steuern, der mit entästeten Baumstämmen beschickt wurde, diese zersägte und die entsprechend zugeschnittenen Abschnitte in

verschiedene Boxen einsortierte.

Die Eingangssignale kamen von Lichtschranken, Endlagenschaltern, Weggebern usw.. Mit den Ausgangssignalen konnte man die Förderer ein- und ausschalten, die Säge hoch- und herunterfahren, Anschlagklappen und Auswerfer bedienen und vieles mehr. Ein übergeordneter Rechner bestimmte über eine fest vorgeschriebene Schnittstelle, wieviel Meter abzuschneiden sind und für welche Box das abgetrennte Stück bestimmt ist. Dieser Rechner übernahm auch die Vermessung des Stamms (bzw. ließ sich die Werte von einem Dritten übergeben).

Zur Beruhigung der Nerven beider Partner wurde ein Pflichtenheft geschrieben, in dem festgehalten wurde, was welches Signal aussagt und wie der technologische Ablauf stattfinden soll. Auch wenn man schon vorher weiß, daß hinterher alles anders aussieht, hat so ein Pflichtenheft einen unschätzbaren Wert.

Simulantentum

Anhand des Pflichtenheftes wurden die Programmoduln entworfen. Die Zuteilung von Aufgabenbereichen zu bestimmten Tasks ist im Großen und Ganzen eigentlich reine Geschmacksache, wobei es allerdings mehr unzweckmäßige als sinnvolle Lösungen gibt.

Um hierbei irgendwelchen abstrakten Entwurfsmethoden zu entrinnen, hat sich die folgende Entwurfsstrategie als besonders amüsant und anschaulich erwiesen:

Man stellt sich vor, daß man viele kleine Männchen hat, die nur einen begrenzten Aufgabenbereich über-

Auch wenn man schon vorher weiß, daß hinterher alles anders aussieht, hat ein Pflichtenheft einen unschätzbaren Wert.

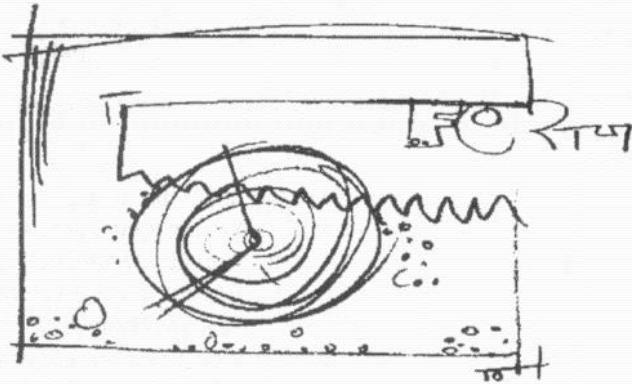
schauen können. Diese Männchen stehen nun an den Schaltstellen auf dem Rundholzplatz und verrichten ihre zugeordneten Jobs. Jedes Männchen wird



im Programm durch eine Task repräsentiert.

Da diese Männchen nicht einfach durcheinanderquasseln dürfen, werden noch Zustände vereinbart, auf die die Jungs zu warten haben, bevor sie zu werke gehen. So bekommt man ohne Entwurfs-HighTech ein bildhaftes Modell zustande, das man auch völlig übermüdet nachts um 3.00 Uhr nicht aus den Augen verliert.

Der Rest der Arbeit bestand aus reinem Codieren. Das Ergebnis war eine schöne Blinkerei der Lampen, wenn man an den (richtigen) Schaltern knipste. Nun konnte auf die mechanische und elektrische Fertigstellung des Sägewerks gewartet werden.



Erster Kontakt zur Realität

Als Programmierer hat man vor kreischenden Sägen und der stillen Kraft von Hydraulikantrieben großen Respekt. Aus diesem Grund wurde eine genaue Strategie der Inbetriebnahme überlegt, um nicht mit einen zu großen Scherbenhaufen zu beginnen.

Aber es kam alles ganz anders. Die Leute vom Sägewerk hatten die Simulationsdiskette schon lange in den Rechner vor Ort gesteckt und gestartet. Sie waren sogar schon in der Lage, irgendwas von einem Baumstamm abzuschneiden, und wußten über verbesserungswürdige Details zu berichten.

Nach dem ersten Schreck konnten Stück für Stück die Anlagenkonstanten (Impulswertigkeiten der Weggeber usw.) bestimmt werden. Allerdings wurden die Arbeiten viel durch chaotische Anlagenzustände unterbrochen. Mal klebte ein Schütz fest, mal wurde eine Lichtschranke durch

die Sonne irritiert, mal sprang das Holz vom Förderer, mal blockierte ein Förderer und und und.

Bei solchen Zuständen war man mit Forth King auf der Anlage, da man ja alle Geräte nach Belieben sofort durch Peripherie-Befehle ein- und ausschalten konnte. An solchen Tagen fiel es einem wirklich schwer, sich vorzustellen, wie man wohl ohne eine interaktiv arbeitende Programmiersprache auskommen sollte.

Feinheiten

Die Tatsachen an der Anlage unterschieden sich naturgemäß gewaltig von dem, was man sich so für die Simulation ausgedacht hatte. Das einzige, was auf Anhieb klappte, war der bruchfreie Notstop der Anlage (...auch nicht ganz unwichtig). Entgegen der Praxis beim Simulieren wurde an der Anlage pausenlos und in allen möglichen und unmöglichen Zuständen

ein Warmstart ausgeführt. Der Kaltstart lief dagegen nur einmal beim ersten Einschalten. Aus diesem Grund mußte der Warmstart das am gründlichsten getestete Stück Programm sein - was er nicht war.

Eine weitere edle Fehlerquelle war das kurze Holz. Es kam selten dort an, wo es sollte.

Meistens sprang es irgendwo vom Förderer oder fiel durch Ritzen

hindurch. Das war immer dann besonders tragisch, wenn ganz emsig auf seine Ankunft gewartet wurde, wie beim Reststück. Nach dem letzten Kappvorgang lief nämlich der Förderer so lange, bis das Reststück die Säge passiert hatte. Danach wurde der nächste Stamm erwartet, der auch schon auf dem Förderer lag. Wenn sich das kurze Holz mal wieder unschön verflüchtigt hatte, wurde der Förderer in Erwartung auf den Ab-

trünnigen immer weiter gedreht. Die unbelechte Software verwechselte den neuen Stamm mit dem Reststück und wollte solange drehen, bis dessen Ende erreicht war. Dabei gab's leider mechanisch großen Ärger, da der Bereich hinter der Säge von den Abmessungen her nicht für ganze Stämme ausgelegt war. Einen Rückwärtsgang gab's auch nicht, so daß das Problem mit einer Motorsäge gelöst werden mußte.

Ein weiteres Problem bereitete die Hebehydraulik der Säge, die nicht ganz dicht war. Nach einiger Zeit senkte sich das wichtigste Stück des ganzen Sägewerks immer weiter ab, so daß es wieder in die obere Endlage zurückgefahren werden mußte. Hierzu mußte hydraulisch bedingt der Heber mit angehoben werden, der dann in den Weg des Stamms hineinragte. Damit hatte man sich ein zum sonstigen Ablauf asynchrones Ereignis eingehandelt, was sich in seiner Vielfalt (auf Grund mangelnder Phantasie) nicht so recht simulieren ließ. So kam es ca. zweimal am Tag vor, daß gerade ein Stamm mit voller Geschwindigkeit in die Säge hineinfuhr, als sich der Heber entschloß, den Weg zu versperren. Das soll fürchterlich gekracht haben und war auch aus mechanischer Sicht nicht besonders vorteilhaft.

Solche unvorhersehbaren Zustände gab's auf diesem Rundholzplatz in Hülle und Fülle. Man könnte mit dem Programmieren sicherlich sein halbes Leben verbringen und würde dann immer noch nicht jeder Situation ge-

In lustigen Stories stecken oft Standardprobleme und ihre Lösungen

wachsen sein. Aus diesem Grund gab's nur eine Überlebensstrategie: Man mußte sich mit den späteren Bedientern bestens verstehen. Auf diese Art ließen sich die Menschen als die vielfältigsten Produzenten unvorhergesehener Ereignisse zu Rettern in Ausnahmeständen verwandeln. Das klappte aber nur deshalb so gut, weil sie es selber wollten. Statt auf das Programm zu meckern, wurde überlegt, wie den Problemen beigegeben

werden konnte. In den meisten Fällen wurde trotzdem das Programm verbessert, was die Bediener wiederum sehr wohlwollend aufnahmen.

Praktisch Forthen

Ein Sägewerk in Betrieb zu nehmen, macht einen Höllenspaß. Um effektiv arbeiten zu können, muß man vorher viel Zeit in den Entwurf stecken und sich in gemütlicher Atmosphäre viel mit Fachleuten über die allgemeine Problematik unterhalten. In lustigen Stories stecken oft Standardprobleme und ihre Lösungen.

Wie man bei einer angemessenen Zeitvorgabe ein Sägewerk ohne Forth in Betrieb nehmen soll, ist mir überhaupt nicht klar. Mit dem standardmäßigen Edit-Compile-Run hat



Schleich- und Eilgang

man schließlich keinen direkten Kontakt zur Anlage. Die Leute vom Sägewerk würden es sicherlich auch etwas witzig finden, wenn man erst einmal ein Programm schreiben müßte, um beispielsweise den Förderer ein Stück weiterlaufen zu lassen. In Forth heißt das so:

```
EILEN KEY DROP HALTEN  
oder im Schneckentempo:  
SCHLEICHEN KEY DROP HAL-  
TEN
```

Nach dem ersten Tastendruck (Enter) geht's los und der zweite läßt alles wieder anhalten. Das erkennen sogar die Sägewerker. Erst EILEN einzutippen und dann bei laufendem Förderer unter Zeitdruck HALTEN eingeben zu wollen, kann man getrost vergessen, da man sich nach McMurphy sowieso vertippt. Das Resultat ist mei-

stens ein entnervter Griff zum Not-stop-Taster.

Als Forthler hat man die (fast) volle Freiheit der Namensgebung. Von kryptischen Kürzeln kann nur dringendst abgeraten werden, da man mit größter Wahrscheinlichkeit auch nach einem viertel Jahr noch in seinem Programm durchsehen muß. Hier helfen die Tips von Leo Brodie aus seinem Buch "In Forth denken" auf die Sprünge. (Kerngedanke: Drücke aus, was das Wort tut und nicht wie es das tut.) Man sollte sich auch davor hüten, die Bezeichnungen von der Anlage in das Programm zu übernehmen. So heißt die Meßlichtschranke in einem Beispielprogramm I.LS11 (I wie Input=Eingangssignal und LS11 als Nummer der Lichtschranke an der Anlage). Witzigerweise steht daneben der aufklärende Kommentar, daß es sich hierbei um die Meßlichtschranke handelt. Zur Lesbarkeit des Programms trägt das aber nicht bei. Wenn man das Ding nun einfach MESS-LS nennt, gibt's aus der Ecke keine Probleme.

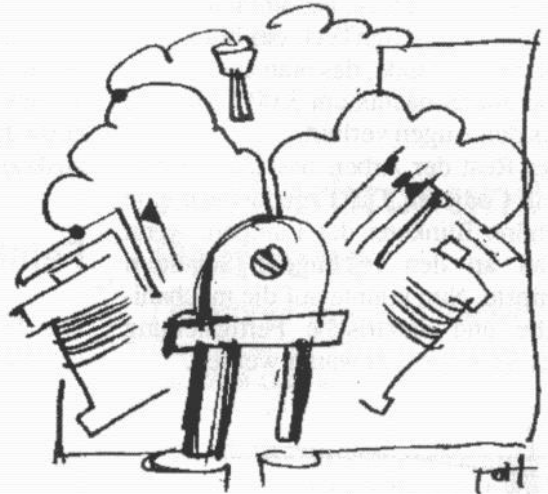
Fazit

Die Programmierung des Sägewerks brachte viele neue Erkenntnisse, die allein die Arbeit schon wert waren. Ein paar als interessant empfundene Gags sollen hier den Abschluß bilden:

- Ein detailliertes Pflichtenheft ist dringend notwendig, auch wenn hinterher vieles anders realisiert wird.
- Die Simulation des Prozesses mit Schaltern und Lampen hilft viele Bugs aus dem Programm zu entfernen. Auf dem Platz hat man aber mit einem Sack voll Überraschungen zu rechnen. Der Entwurf und die Simulation waren gut, wenn das Programmkonzept erhalten bleiben kann.
- Die ersten Tage an einer neuen Anlage verpläpert man zum großen Teil mit Warten, weil laufend irgendetwas aussteigt, auf das man

keinen Einfluß hat, was das Testen des Programms aber unmöglich macht.

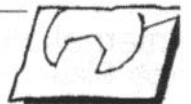
- Man nimmt nach einer Programmänderung auch die alte Version



„Besonders gefährlich sind Elektriker, die mit Relais die tollsten Sachen verkabeln.“

noch einmal mit, da in der Zwischenzeit Elektriker und Mechaniker an der Anlage herumbasteln und z. B. vergessen, irgendwelche Drähte wieder anzuklemmen.

- Besonders gefährlich sind Elektriker, die mit Relais die tollsten Sachen verkabeln, wie Anlaufsperrern oder langsames Anfahren und Stoppen der Förderer. Mit ihnen muß man sich viel unterhalten, einerseits über das, was sie gerade tun, und andererseits, was für neue Ideen sie haben. So kann man sich viele Rätseleien über "merkwürdige Effekte" ersparen.
- Asynchron zum Prozeß laufende Ereignisse sorgen kräftig für Würze. Programmteile, die davon betroffen sind (in diesem Fall die Task, die die Säge hochziehen hatte) werden schnell größer, um die unerfreulichen Nebeneffekte zu beherrschen.
- Wo im Sägewerk kleines Holz umherpurzelt, ist nichts unmöglich.
- Der spätere Anlagenfahrer hat die Schlüsselrolle beim Gelingen des Projekts. Wenn man sich mit ihm nicht gut versteht, hat man verloren.



Internationaler Zeichensatz im F-PC

von Klaus Vogt

Ein wirklich häßlicher Umstand beim ansonsten sehr schönen F-PC ist die Tatsache, daß Umlaute und Sonderzeichen im Editor nur mit großen Bocksprüngen eingegeben werden können. Wie dieses Problem beseitigt werden kann, wird beschrieben.

Da ich im Rahmen meiner Diplomarbeit das F-PC-Hypertextsystem in eine mittelgroße Applikation einbinden wollte, schien mir das nicht mehr tragbar. Das vorgestellte Programm unterstützt neben der Eingabe in Editor und Zeileneditor auch das korrekte Suchen, Klein-Groß-Konvertieren und Wortgrenzen-Suchen. Ein Fehler im F-PC führt bei Eingabe von Zeichen 127 innerhalb von Namen (in bisher sehr seltenen Konstellationen) zum Absturz. Daher wird auch die volle Unterstützung von Grafik- und Sonderzeichen in Namen ermöglicht, wozu das Speicherformat von Namen (analog volksForth) verändert wird.

Der internationale Zeichensatz ...

... war sicherlich eins der wenigen Glanzlichter, die IBM bei Entwicklung seines PCs zu bieten hatte. Manch eingefleischter UNIX-Fan schaut neiderfüllt auf die vergleichsweise harmlosen Probleme, die ein 'ö' oder 'ß' einem PC verursacht. Zwar liegen sie auch beim PC separat (außerhalb des normalen Zeichensatzes der Codes 0..127) und werden daher auch von kaum einem Compiler (ausgenommen natürlich volksFORTH) wie normale Zeichen behandelt. Die meisten Editoren kommen

aber problemlos damit klar; der vom F-PC leider nicht.

IBM mußte sich entschließen, um mehr als 256 verschiedene Tastenkombinationen unterscheiden zu können, einige als zwei-byte-Codes (eine vorangestellte Null) darzustellen.

Der Rückschritt des F-PC

Tom Zimmer gefiel das offenbar nicht. Seine Tastatur-Einleseroutine quetscht diese zwei-byte-Codes nunmehr wiederum in den bereits belegten Bereich (128 bis 255) zurück. Dies erleichtert vielleicht die Verwaltung von Tastentabellen, führt aber andererseits dazu, daß beispielsweise

Listing zu: Internationaler Zeichensatz im F-PC von Claus Vogt

```

\\ keyb.seq : International keyboard for F-PC.  Claus Vogt 1992

Supporting the international and graphical charset in Editor and Forth-Words.

If you want to put the changes into your kernel, just a few lines
are changed. Most of the code is a copy of words which are patched.

The third part of this file changes the format names are placed
into memory. You may comment it out, but read the comment about
the problems you will have if you do so (you already might have
them if you ever turned CAPS OFF because of an error inside F-PC)

Claus Vogt (Berlin) 1992. Mitglied der Forth-Gesellschaft.

tested with F-PC 3.56, few portability problems expected for other versions

( \ changes to editor and line-editor, BIOSKEY and BIOSKEY?

only forth also definitions decimal

\ - \warn      : \warn  warning @ 0= IF [compile] \ THEN ; immediate
                \ do line if WARNING ON

: patch! ( cfa1 cfa2 -- )          \ patches cfa2, so that it executes cfa1.
  save!> dp [ Assembler ]        \ works on : and CODE -definitions
  >pre >r MOV AX, # r> A; JMP AX a; pre> \ cfa1 is lost: 5 bytes overwritten
  [ hidden ] restore> dp ;

hidden definitions

CODE oldBIOSKEY? ( -- f1 )        \ exactly the same as old BIOSKEY?
\ Return a true flag if a key, other than control break, has been pressed.
  BEGIN
    MOV AH, # 1
    PUSH SI          PUSH BP
    INT $16
    POP BP           POP SI
    MOV BIOSCHAR AX
  
```

Stichworte

F-PC
 patchen
 Sonderzeichen
 uppercase

Zum Autor:

Claus Vogt

(Bülowstr. 67, W-1000 Berlin 30) Jahrgang 1960, ledig, ein Kind, ist Programmierer am Biomechanik-Labor des FU-Klinikum Steglitz und Teilzeitstudent des Maschinenbaus an der TU Berlin. Seine ersten Schritte in Forth erfolgten in ultraFORTH, das er vom C64 auf C16 portierte. Derzeit programmiert er Meß-, Steuer- und Grafiksoftware in F-PC.



ein 'ö' denselben Code wie ein Alt-T liefert. Jedes im Editor versehentlich gedrückte 'ö' wird mit einer hübschen kleinen Alert-Box quittiert, die 'Alt-T' verstanden hat und daher die Tabellen-sprünge auf irgendwelche sinnlosen Werte setzt. Dies konnte ich akzeptieren, solange es nur um die Eingabe von Quelltexten ging. Seit ich den Editor zur Einbindung eines Hypertext-Hilfesystems in eine größere Applikation benutze, schien es mir untragbar.

Die Lösung für Editor und Zeileneditor

Die vorgestellte Lösung beruht auf einer einfachen Idee. Wo IBM zwei bytes hatte und Tom Zimmer eins draus machte, mache ich wieder zwei draus. Allen Zeichen, für die IBM Codes zwischen 128 und 255 verwendet, stellt das verwendete Programm ein Präfix von 128 voran. Somit erscheint ein gedrücktes 'ö' nunmehr als zwei gedrückte Tasten, einem Präfix 128 und dem üblichen Code 148. Im Editor führe ich eine neue Funktion ein (das geht erstaunlich einfach), die auf dem Tastencode 128 (im F-PC Alt-9) liegt und eine weitere Taste liest und in den Text einfügt. Diese Funktion braucht praktisch nie benutzt zu werden, das ganze ist für die Benutzung transparent. In identischer Art wird die entsprechende Zeileneditorfunktion erzeugt und installiert.

Für die Installation der neuen Tastatur-Leseroutine wurde BIOSKEY und BIOSKEY? gewählt, da dadurch vermutlich die geringsten Kompatibilitätsprobleme auftreten dürften und auch eine ev. Einbindung ins F-PC mit geringstem Reibungsverlust abgehen dürfte. Diese zwei Worte müssen gepatched werden, hierfür habe ich wieder einmal eine neue Variante benutzt, die den Assembler anwirft, um den alten CFA mit einem: MOV AX, # NEWCFA, JMP AX zu überschreiben. Alle, die solche Tricks mögen, werden über die Quelle köstlich lachen. Inkompatibilitäten mit bestehenden Programmen sind zu erwarten, wenn ein Programm die Taste 128 (ein C-Cedille: Ç) speziell behandelt. Die

Fortsetzung des Listings zu: Internationaler Zeichensatz

```

0= IF          \ ??? from where is this ZeroFlag ??? /clv
    MOV AX, # 0
    1PUSH
    THEN
    CMP AX, # 0 \ Ignore Control Break keys
0= WHILE
    MOV AH, # 0 \ That is, throw them away
    PUSH SI     PUSH BP
    INT $16
    POP BP      POP SI
    REPEAT
    MOV AX, # -1
    1PUSH       END-CODE
CODE oldBIOSKEY ( -- c1 ) \ exactly the same as old BIOSKEY
\ Return the value of the next key, other than control break.
    BEGIN
    MOV AH, # 0
    PUSH SI     PUSH BP
    INT $16
    POP BP      POP SI
    CMP AX, # 0 \ Ignore Control BREAK, 00 Hex.
0<> UNTIL
    MOV BIOSKEYVAL AX
    1PUSH       END-CODE

0 Value 2ndChar

: newBIOSKEY ( -- c ) \ will be patched upon old BIOSKEY
  2ndChar \ waiting char?
  IF 2ndChar dup BIOSKEYVAL ! off> 2ndChar EXIT THEN
  oldBIOSKEY dup $0080 and 0= ?exit \ not 128..255
  =: 2ndChar \ has to wait
  $80 ; \ 1st char is 128

: newBIOSKEY? ( -- flag ) \ will be patched upon old BIOSKEY?
  2ndChar 0<> ?dup ?exit oldBIOSKEY? ;

' newBIOSKEY ' BIOSKEY patch!
' newBIOSKEY? ' BIOSKEY? patch!

\warn cr .( .. BIOSKEY and BIOSKEY? patched!)

\ new editor function
editor also definitions

: interschr ( -- ) key schr ; \ enter 2ndchar to editor

128 fnset interschr

\warn cr .( .. in Editor Alt-9 redefined input special char )

previous forth definitions

\ new lineditor function
hidden also definitions

: inter<ichar> key <ichar> ;

keyfuncptr
  >keys1 ' inter<ichar> 128 lkey!
  >keys2 ' inter<ichar> 128 lkey!
=: keyfuncptr

\warn cr .( .. in Lineditor Alt-9 redefined to input special char )

previous forth definitions

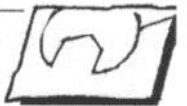
}

( \ Editor changes to support word-skipping

\ the following may cause problems because the word-skipping
\ till yet treated all chars>127 as blanks

editor also definitions
\ patching editor word-selection-subfunctions
\ all the same but now treating Ascii 128 .. 255 as nonblank

: new>space ( --- ) \ move to next space in line
  linelen dup screenchar over min
    
```

Fortsetzung des Listings zu: Internationaler Zeichensatz

```

?do   linebuf 1+ i + c@ dup bl =
\     swap 127 > or
      NIP
      if     drop i leave then
loop  =: screenchar ;

: new<<space> ( --- ) \ move to non blank in line
linelen dup screenchar over min
?do   linebuf 1+ i + c@ dup bl <>
\     swap 127 > 0= and
      NIP
      if     drop i leave then
loop  linelen min =: screenchar ;

: new<<space> ( --- nl ) \ nl = offset from line strt to prev space
0 dup screenchar
?do   linebuf 1+ i + c@ dup bl =
\     swap 127 > or
      NIP
      if     drop i leave then
-1 +loop dup =: screenchar ;

: new<text> ( --- ) \ move to previous text in line.
0 dup screenchar
?do   linebuf 1+ i + c@ dup bl <>
\     swap 127 > 0= and
      NIP
      if     drop i leave then
-1 +loop =: screenchar ;

hidden
' new<<space> ' <<space> patch!
' new>>space ' >>space patch!
' new>>space ' >>space patch!
' new<text> ' <text patch!

previous forth definitions
)

( \ new Casetranslation for Editor-search and system

\ the following may cause problems because till yet all
\ chars>127 where converted to those <128 when converted to upper

: c!+ ( addr 8b -- addr+1 ) over c! 1+ ;

ATBL \ change uppercase translation to handle international chars
\ now chars from 127 to 255 are no longer translated to 0..127
\ characters below 128 are the same as now
0 c!+ 1 c!+ 2 c!+ 3 c!+ 4 c!+ 5 c!+ 6 c!+ 7 c!+
8 c!+ 32 c!+ 10 c!+ 11 c!+ 12 c!+ 13 c!+ 14 c!+ 15 c!+
16 c!+ 17 c!+ 18 c!+ 19 c!+ 20 c!+ 21 c!+ 22 c!+ 23 c!+
24 c!+ 25 c!+ 26 c!+ 27 c!+ 28 c!+ 29 c!+ 30 c!+ 31 c!+
32 c!+ '!' c!+ '' c!+ '#' c!+ '$' c!+ '%' c!+ '&' c!+ ''' c!+
'(' c!+ ')' c!+ '*' c!+ '+' c!+ ',' c!+ '-' c!+ '.' c!+ '/' c!+
'0' c!+ '1' c!+ '2' c!+ '3' c!+ '4' c!+ '5' c!+ '6' c!+ '7' c!+
'8' c!+ '9' c!+ ':' c!+ ';' c!+ '<' c!+ '=' c!+ '>' c!+ '?' c!+
'@' c!+ 'A' c!+ 'B' c!+ 'C' c!+ 'D' c!+ 'E' c!+ 'F' c!+ 'G' c!+
'H' c!+ 'I' c!+ 'J' c!+ 'K' c!+ 'L' c!+ 'M' c!+ 'N' c!+ 'O' c!+
'P' c!+ 'Q' c!+ 'R' c!+ 'S' c!+ 'T' c!+ 'U' c!+ 'V' c!+ 'W' c!+
'X' c!+ 'Y' c!+ 'Z' c!+ '[' c!+ '\' c!+ ']' c!+ '^' c!+
_ c!+ 'A' c!+ 'B' c!+ 'C' c!+ 'D' c!+ 'E' c!+ 'F' c!+ 'G' c!+
'H' c!+ 'I' c!+ 'J' c!+ 'K' c!+ 'L' c!+ 'M' c!+ 'N' c!+ 'O' c!+
'P' c!+ 'Q' c!+ 'R' c!+ 'S' c!+ 'T' c!+ 'U' c!+ 'V' c!+ 'W' c!+
'X' c!+ 'Y' c!+ 'Z' c!+ '[' c!+ '\' c!+ ']' c!+ '^' c!+ 'W' c!+

\ Characters above 127 are no longer (!) translated to below 127

$80 c!+ 'U' c!+ 'E' c!+ $83 c!+ 'A' c!+ $85 c!+ 'A' c!+ 'Ç' c!+
$88 c!+ $89 c!+ $8A c!+ $8B c!+ $8C c!+ $8D c!+ $8E c!+ $8F c!+
$90 c!+ $91 c!+ $92 c!+ $93 c!+ 'O' c!+ 'U' c!+ $96 c!+ $97 c!+
$98 c!+ $99 c!+ $9A c!+ $9B c!+ 'N' c!+ $9D c!+ $9E c!+ $9F c!+
$A0 c!+ $A1 c!+ $A2 c!+ $A3 c!+ $A4 c!+ $A5 c!+ $A6 c!+ $A7 c!+
$A8 c!+ $A9 c!+ $AA c!+ $AB c!+ $AC c!+ $AD c!+ $AE c!+ $AF c!+
$B0 c!+ $B1 c!+ $B2 c!+ $B3 c!+ $B4 c!+ $B5 c!+ $B6 c!+ $B7 c!+
$B8 c!+ $B9 c!+ $BA c!+ $BB c!+ $BC c!+ $BD c!+ $BE c!+ $BF c!+
$C0 c!+ $C1 c!+ $C2 c!+ $C3 c!+ $C4 c!+ $C5 c!+ $C6 c!+ $C7 c!+
$C8 c!+ $C9 c!+ $CA c!+ $CB c!+ $CC c!+ $CD c!+ $CE c!+ $CF c!+
$D0 c!+ $D1 c!+ $D2 c!+ $D3 c!+ $D4 c!+ $D5 c!+ $D6 c!+ $D7 c!+
$D8 c!+ $D9 c!+ $DA c!+ $DB c!+ $DC c!+ $DD c!+ $DE c!+ $DF c!+
$E0 c!+ $E1 c!+ $E2 c!+ $E3 c!+ $E4 c!+ $E5 c!+ $E6 c!+ $E7 c!+
$E8 c!+ $E9 c!+ $EA c!+ $EB c!+ $EC c!+ $ED c!+ $EE c!+ $EF c!+
$F0 c!+ $F1 c!+ $F2 c!+ $F3 c!+ $F4 c!+ $F5 c!+ $F6 c!+ $F7 c!+

```

französischen Nachbarn brauchen nichts zu befürchten, da auch dieser hübsche Buchstabe durch ein vorangestelltes (identisches) Präfix geschützt wird. Eine Veränderung von KEY und KEY? ist unproblematisch, solange sie weiterhin auf BIOSKEY und BIOSKEY? aufsetzen, was die zahlreichen im F-PC vorhandenen KEYS und KEY?s offenbar alle tun.

Das Suchen von Wortanfängen und -enden...

...ist nicht nur für die Cursorbewegung sondern auch für den Textumbruch und das separieren von Hyper-textverweisen wirksam. Editorfunktionen wurde verändert, um auch hier Umlaute wie jedes amerikanische Zeichen verwenden zu können. Bisher wurde im Editor jedes Zeichen über Ascii 127 pauschal als Leerzeichen betrachtet. Dies ist jetzt nicht mehr der Fall, hieraus folgende Inkompatibilitäten habe ich bisher nicht beobachten können. Sie sind aber durchaus denkbar.

Die Konversion von Klein- in Großbuchstaben...

... bewirkte im F-PC bisher eine Abbildung aller Zeichen 128..255 auf die Zeichen 0..127. Insbesondere, um auch ein komfortables Suchen im Editor zu ermöglichen wurde die Konversions-Tabelle ATBL verändert, wobei ich auch alle ausländischen Umlaute, die ich spontan identifizieren konnte, entsprechend eintrug. Die übrigen (Grafik-)Zeichen werden nicht verändert. Durch diese Änderung verhalten sich UPC UPPER und ?UPPERCASE anders, wodurch ein Fehler im F-PC sichtbar wurde. Weitere Inkompatibilitäten sind denkbar, aber eher unwahrscheinlich.

F-PC Fehler bei Sonderzeichen im Namen

Wenn im F-PC bisher CAPS OFF gesetzt wurde, wurden Zeichen 127 im Namensfeld abgelegt. Dies ist eindeu-

tig ein Fehler, da das vom F83 übernommene Format des Namensfeld das oberste Bit ausschließlich zum Erkennen des Namensende benutzt. Je nach Kontext stürzt das System sofort oder später bei einer bestimmten Eingabe ab. Dieses Problem ist im Quelltext näher erläutert. Da die Wahrscheinlichkeit dieses Absturzes durch das vorgestellte Programm erheblich erhöht wird, entschloß ich mich, eine allgemeine Lösung zu implementieren, die sich nebenbei auch als die einfachste und fehlertoleranteste herausstellte.

Sonderzeichen innerhalb von Namen werden voll unterstützt.

Das vom F83 übernommene Namensformat mit dem gesetzten achten Bit als Endezeichen wird nicht weiter verwendet. Die dazu nötigen Änderungen beziehen sich in erster Linie auf das Wort (FIND), das aufgrund seiner Auslösefunktion für obigen Fehler ohnehin geändert werden mußte. Weitere kleinere Änderungen finden sich in "HEADER NAME ALIAS und %.ID". Außerdem sind selbstverständlich sämtliche bereits im System vorhandenen Namen auf das neue Format umzustellen. Inwieweit diese tiefgreifende Änderung Inkompatibilitäten auslösen kann, ist läßt sich nur schwer abschätzen. Da allerdings nur wenige Worte unter Umgehung der o.g. Worte auf das Namensfeld zugreifen, dürften selten Probleme auftreten. Ohnehin erfordern alle übernommenen F-83-Programme, die dies tun, Änderungen bei der Portierung auf F-PC, da hier die Namen in einem gesonderten Segment abgespeichert werden.

Erste Erfahrungen...

... waren durchwegs positiv. Die Möglichkeit, Umlaute in Hypertext-Verweisen und im Editor benutzen zu können, habe ich reichlich benutzt und kann sie als gut getestet bezeichnen. Sonderzeichen innerhalb von Namen verwende ich schon aus Gründen der Portabilität nur ungern, so daß

die Folgen dieser Option sicherlich nicht mich treffen werden. Alle im Editor belegten Tasten wie die oben erwähnte Taste Alt-T können bei der vorgestellten Lösung wei-

terhin benutzt werden. Eine gewisse Umstellung ist für jene erforderlich, die bisher die Tab-Sprünge mit 'ö' verstellt haben.

Fortsetzung des Listings zu: Internationaler Zeichensatz

```

$FB c!+ $F9 c!+ $FA c!+ $FB c!+ $FC c!+ $FD c!+ $FE c!+ $FF c!+
drop
CODE newCAPS-COMP ( addr1 addr2 len -- -1 | 0 | 1 )
\ Perform a comparison of two strings, but ignore Case differences.
  MOV DX, SI      POP CX
  POP DI         POP SI
  PUSH ES       MOV ES, SSEG
  mov BX, # atbl          \ added/clv
  BEGIN
    JCXZ 0 $
  \   MOV AH, 0 [SI]      INC SI      \ removed/clv
  \   MOV ES: AL, 0 [DI]  INC DI      \ "
  \   OR AX, # $02020    CMP AH, AL  \ "
  \   mov AL, 0 [SI]     xlat inc SI  mov AH, AL \ inserted/clv
  \   mov ES: AL, 0 [DI] xlat inc DI  cmp AH, AL \ "
    JNE 1 $          DEC CX
  AGAIN
  1 $: 0< IF
    MOV CX, # -1
  ELSE
    MOV CX, # 1
  THEN
  0 $: MOV SI, DX
    POP ES
    PUSH CX
    NEXT          END-CODE
' newCAPS-COMP ' CAPS-COMP patch!
)
( \ making F-PC accepting chars >127 in names
\ The following may cause problems because the format of laying
\ down names by F-PC is changed. Many few words act on this format
\ directly, so the problems may not be so hard.
\ And I think they are even smaller than if you ever turned CAPS OFF..
\ ***** The error of turning CAPS OFF: *****
\ There is an error in (FIND) and the way F-PC (3.54 and 3.56) handles
\ headers when
\ you use names with chars >127 inside a name:
\ Try: CAPS OFF : xxyy ; ' xxyy U. ' xxlyy U.
\ ( Use ALT-A to enter char 177 ).
\ The first ' returns the correct value, the 2nd one returns
\ the 'yy' inside the name as cfa. So the system will crash if
\ you execute xxlyy. This is because HEADER layed down the chars
\ as received ( with 7th bit turned on) because CAPS OFF (and
\ my new ATBL above too) does'nt let ?UPPERCASE remove it.
\ (FIND) finds the names xxyy and xxlyy to be equal because the first char
\ differing, just differs in the 7th bit and this is also the point
\ for (FIND) to use as end-of-name-flag and returning the following
\ word as cfa. (Other crashes are possible: CAPS OFF : HELLO ;
\ crashes my system F-PC3.54 just after this line. I don't know why.)
\ I decided to change the format of the header. Now the 7th bit is
\ no longer set. Any char is a legal name-char. I hope that I found
\ all words using this bit but I'm not truely sure:
: new"HEADER ( str-addr -- )
  SPCHECK
  DUP C@ 31 > ABORT" Name TOO LONG, > 31 chars!"
  ?ALREADY_DEF
  ALIGN YHERE 2- Y@ CNHASH HERE CNHASH <>
  IF YHERE HERE CNHASH DUP Y@ ROT UMIN SWAP
    Y! ( >NAME hash entry )
  THEN ,VIEW
  YHERE OVER CURRENT @ HASH DUP @ Y, ( link ) ! ( current )
  YHERE LAST ! ( remember nfa )
  YHERE ?CS: ROT DUP C@ WIDTH @ MIN 1+ >R ( yh cs str )
  YHERE YS: R@ CMOVEL ( copy str ) R> YDP +! ALIGN (
nam )
```



Fortsetzung des Listings zu: Internationaler Zeichensatz

```

128 SWAP YCSET
\ 128 YHERE 1- YCSET ( delimiter Bits )
HERE Y, ( CFA in header )
YHERE HERE CNHASH 2+ Y! ( valid stopper in next n hash entry )
;

: newALIAS ( A1 | alias_NAME --- )
\ HEADER LAST @ 1 TRAVERSE 1+ Y! ;
HEADER LAST @ ycount $3f and + Y! ;

: newName> ( anf -- acf )
\ 1 TRAVERSE 1+ Y@ ;
ycount $3f and + Y@ ;
: newTRAVERSE true abort" new Name format installed by KEYB.SEQ" ;

: new*.ID ( nfa -- )
\ Display the variable length name whose name field address is on the
\ stack.
\ The old version does: If it is shorter than its count, it is padded with underscores.
\ Only valid Ascii is typed.
\ The new version types all chars 0..255
\ DUP 1+ DUP YC@ ROT YC@ 31 AND 0 \ why used it 31, not $3f??
\ ?DO DUP 127 AND FEMIT 128 AND
\ IF ASCII _ 128 OR ELSE 1+ DUP YC@ THEN
\ LOOP 2DROP SPACE ;
ycount $3f and bounds ?DO I yc@ femit LOOP space ;

CODE new(FIND) ( here alf -- cfa flag | here false )
POP DI
OR DI, DI
0= IF \ alf=0 : end of list
PUSH DI
NEXT

THEN
POP BX \ DS:BX = here
PUSH ES PUSH SI
MOV ES, YSEG \ ES:DI=alf
BEGIN
MOV ES: CX, 2 [DI]
AND CL, # $3f \ mask immediate bit
CMP CX, 0 [BX]
0= IF \ count and first char equals
XOR CH, CH \ CX=count
MOV SI, BX MOV DX, DI \ DS:SI=here
ADD DI, # 4 ADD SI, # 2
DEC CX \ first char already done ZF=1 if CX=0!
CLD
REPE CMPSB
0= IF \ strings match
MOV ES: CX, 0 [DI] \ pick up CFA
MOV DI, DX
XOR AH, AH
MOV ES: AL, 2 [DI]
AND AL, # 64
0<> IF
INC AX \ AX=1
ELSE
DEC AX \ AX=-1
THEN
POP SI
POP ES
PUSH CX
1PUSH

THEN
MOV DI, DX
MOV SI, BX

THEN
MOV ES: DI, 0 [DI]
OR DI, DI
0= UNTIL \ end of list
POP SI POP ES
PUSH BX
PUSH DI \ =0
next END-CODE

\ Create save(find) 5 allot ' (find) save(find) 5 cmove
\
\ : t ( ;i name -- ) \ output of DEFINED name of old and new (FIND)
>in @ cr defined u. u. >in !
\ ['] new(find) ['] (find) patch!
\ defined u. u.
\ save(find) ['] (find) 5 cmove ;
\

```

Glossar

```

patch! ( cfa1 cfa2 -- )
patches cfa2 so that it executes
same as cfa1. This kind of patching
may be used on most kinds of
definitions including Code-definitions.
CFA2 is replaced by: MOV AX, # cfa1
JMP AX this uses 5 bytes which could be a
problem with CREATE DOES> -constructs
without any body. This word is highly
unportable!
cr ( .. sample for patch! : )
code a mov ax, # $a 1push
end-code
code b mov ax, # $b 1push
end-code
: c $c ;
: d $d ;
' c ' a patch! cr
. ( Code: A now executes
Colon: C )
' b ' d patch! cr
. ( Colon: D now executes
Code: B )
\ warn ( ;i line -- )
\ conditionally warning-actions
If WARNING is ON the following
line is treated as normal Forth-Code.
If WARNING is OFF the line is treated
as comment.
C!+ ( addr 8b -- addr+ )
Stores 8b at addr and returns addr
after it. Useful to change arrays,
but remark that parameter are
transferred in reverse order than
with C! .
oldBIOSKEY ( -- char )
oldBIOSKEY? ( -- flag )
An exact copy of the old BIOS-KEY
and BIOSKEY? ( of F-PC 3.56 ).
newBIOSKEY ( -- char )
newBIOSKEY? ( -- flag )
New words replacing the old versions.
They simulate in case of Input of
chars between 128 and 255 the Input
of two consecutive chars: A prefix
of 128 preceding the original key
pressed.
interSchr ( -- ) Editor-Function
inter<ichar> ( -- ) Lineeditor-Function
A new Function allowing to enter
any char (including control- and
graphical-char) to be included in
the Text after pressing the Alt-9-
key ( Ascii 128 ).
new<<space>>
new>space
newspace>
new<text>
New words replacing the old editor-
words for searching of word-
boundaries.
newCAPS-COMP ( addr1 addr2
len -- -1 | 0 | 1 )
New word replacing the old CAPS-COMP
Perform a comparison of two strings,
but ignore Case differences. Uses
ATBL to translate lower to upper,
which old version does not.

```


Fortsetzung des Listings zu: Internationaler Zeichensatz

```
: word>7b ( nfa -- )
  0 0 at
  ycount $3f and + 1- dup yc@ $7f and swap yc! ;

: words>7b ( ;i pars.. -- ) \ reset 7th bit of names of all words
\warn cr ." .. changing all names to have last char 7-bit-wide"
\warn cr ." .. DON'T stop it!!! System will crash"
[' word>7b save!> .id
at? 0 0 at
contextonly . >in @ . #tib @ .
words
at
restore> .id
[' new(find) ['] (find) patch!
[' new(header) ['] "header patch!
[' new(ALIAS) ['] ALIAS patch!
[' new(NAME) ['] NAME> patch!
[' new(TRAVERSE) ['] TRAVERSE patch!
[' new(%id) ['] %id patch!
\warn cr ." .. done"
;
\warn cr ." .. replacing;"
\warn cr ." ( "HEADER (FIND TRAVERSE NAME> ALIAS %ID )

words>7b *.*
forget word>7b

cr ." .. KEYB installed:full support of )
cr ." ( chars 0..255 in Editor and Names )

: .KEYB ( --- )
  defers .hello
  CR TAB
  ." \l International KEYBoard"
  ." \l and graphical characters supported"
  CR ;
' .KEYB IS .hello
}
```

Ergänzende Hinweise

Zum Problem der deutschen Umlaute und Sonderzeichen im F-PC gibt es einige Lösungen, auf die in der VD hingewiesen wurde. *Arndt Klingelberg* erwähnt eine vollständige Lösung und entsprechende Änderungen bei der Vorstellung seines F-PC-ak (Sept.1990, S.9 u. Dez. 1990, S.6). Eine elegante und sehr forth-gemäße Lösung bietet *Ulrich Hoffmann* in der VD März 1991, S.12 in dem Artikel F-PC spricht deutsch an. Unter GERMAN.SEQ wird der Quellcode z.B. von *Jörg Staben* zu dem von ihm vertriebenen F-PC mitgeliefert. Auch *Tom Zimmer* hat sich auf Anregung von *Andreas Goyke* des Themas angenommen. Eine echte Lösung bietet F-PC (ab V. 3.5606) aber dennoch nicht. Es wird lediglich die Eingabe der Umlaute und des Eszett (ß) erleichtert. Die *Vogt'sche* Lösung erweitert die Palette der untereinander nicht kompatiblen F-PC-Derivate. rk

Glossar

```
new%.ID ( nfa -- )
new"HEADER ( str-addr -- )
newALIAS ( Al | ali-
as_NAME --- )
newNAME ( anf -- acf )
new(FIND) ( here alf --
cfa flag
| here false )
```

New words replacing the old versions to support characters 127 inside forth-names. Bit 8 is no longer set as a mark of the end of the name. Instead of this the count byte is used to walk across. The biggest changes occurred to (FIND) the other are just a few bytes shorter.

newTRAVERSE

New word replacing the old version. It aborts with an error-message, so that loading of many (not all) programs using the name-fields directly will cause an error to remark the adaptations being necessary.

```
word7b ( nfa -- )
words7b ( ;i pars.. -- )
  Temporarily used words to convert all names present in your system to the new memory-format, by resetting the highest bit of the last char.
```

FORTH ENTWICKLUNGSUMGEBUNG

**Modell TDS2020
16 Bit, 20 MHz CPU H8/532 Hitachi
Starter Pack**

TDS2020-PIN:

Computerboard mit H8/532 CPU
auch geeignet für direkten Anschluß an Tastatur
mit 64 Keys und LCD-Display.
(Größe: 100 x 80 mm)

TDS2020-DV

Piggyback Entwicklungsboard mit Forth.

TDS2020-PA:

Adapterboard zur Programmierung von H8
EPROM.

DS1213C:

Batteriegepufferter Sockel für S-RAM.

TDS-PC:

Entwicklungssoftware für IBM-PC mit Applika-
tion-Library. 1 Jahr Update-Service.

Handbücher:

Hitachi Hardware Manual,
Hitachi Programming Manual,
TDS2020 Technical Manual.

Komplett Preis:

DM 930,-- + MWSt.

Einführungspreis:

DM 795,-- + MWSt.

Zusätzlich erhältlich:

Datalogger Modul TDS2020-CM
mit PCMCIA Memory Card.

Lascar Electronics Produktions- und VertriebsgmbH

Vordere Kirchstraße 4, D-7241 Eutingen-2
Telefon: 0 74 59 / 12 71, Telefax: 0 74 59 / 24 71

Fourieranalyse in FORTH

von Klaus Kohl

Viele Leute haben mich schon gefragt, ob ich nicht Routinen für eine FFT-Analyse habe. Dieser Artikel soll deshalb mit dem Listing und einem Beispiel diesen Nachfragen abhelfen.

Die Kontrollstruktur FOR . . . NEXT ist deshalb in diesem Beispiel, weil das Programm normalerweise auf dem FORTH-Prozessor NC4000P läuft. Eine Anpassung an volksFORTH ist angegeben. Ein Nachteil des NC4000P ist, daß er Schwierigkeiten bei vorzeichenbehafteter 16-Bit-Arithmetik hat. Deshalb wurden alle Skalierungen so gewählt, daß bei Multiplikation und Division der Absolutbetrag der Werte im Bereich von 0 bis 16383 liegen.

Da das Programm bis auf die Datenfelder der Signale keine Werte speichert, mußten Routinen für Sinusberechnung und Quadratwurzelermittlung implementiert werden. Da sie unabhängig von der FFT-Analyse verwendbar sind, wurde ihnen auch etwas mehr Aufmerksamkeit gewidmet.

Sinus und Cosinusberechnung in FORTH

Natürlich wird der Sinus über eine Reihenentwicklung ermittelt. In einer älteren ELEKTRONIK Ausgabe (7/84) sah ich vor vielen Jahren einmal die Formel:

$$\sin x = 0.9997x - 0.165629x^3 + 0.0074886x^5$$

Diese Formel soll einen relativen Fehler kleiner als 2^{-15} im ersten Quadranten besitzen und der Winkel wird in

rad angegeben. Durch geeignete Umformung entstand die im Screen 4 an-

gegebene Formel, in der ein 16-Bit-Wert (0=0 Grad; 16384=90 Grad ...) in seinen Sinuswert umgewandelt und mit 16384 multipliziert wird. Um Schwierigkeiten in der FFT-Analyse aus dem Weg zu gehen, wird der größte Wert auf 16383 (statt 16384) heruntergesetzt.

Ich habe noch keine ausführlichen Test's mit der Formel und den Ergebnissen durchgeführt, jedoch glaube ich (schon wegen der geringen Fehler in der FFT), daß er unter 2 Bit liegt.

Der Cosinus ist nur ein verschobener Sinus und deshalb muß nur 16384 zum Winkel-Wert addiert werden.

Fourieranalyse - Was ist das? Wer braucht sowas?

Ein Erklärungsversuch mit Kurzkrimi

...

Der Kommissar hatte von Anfang an den Verdacht, daß die drei Steinmetze ihren Kollegen deckten. Aber wenn keiner einen Fehler machte, war das Alibi nicht zu erschüttern. Zwar existierte zufällig eine Tonbandaufnahme, auf der deutlich zu hören war, daß mehrere Steinmetze ihrer schweißtreibenden Tätigkeit nachgingen, doch ob es nun drei oder vier waren, konnte man beim besten Willen nicht heraushören. Zu unregelmäßig waren die Geräusche der klingelnden Meißel. Zu ähnlich die Schlagzahlen der Männer. Mal hörte man die Schläge wie das unregelmäßige Prasseln des Hagels. Dann wieder kamen die Schläge gleichmäßig in kurzer Folge hintereinander wie das Stakato eines Schlagzeugers. "Haben Sie das Band schon ins Labor gegeben?" fragte der junge Assistent. . . .

Genug. Nicht jeder Forthler ist unbedingt auch Krimi-Fan. Dennoch soll jetzt

verraten werden, daß die Leute im Labor zweifelsfrei nachweisen konnten, daß auf der Tonbandaufnahme nur drei Steinmetze bei der Arbeit zu hören waren. Aufschluß gab die Anwendung der Fourier-Analyse (nach dem französischen Mathematiker Jean Baptiste Fourier, 1768 bis 1830), auch Harmonische Analyse oder

ganz wissenschaftlich Mittlere quadratische Approximation durch goniometrische Funktionen. Einfach ausgedrückt wird bei diesem Verfahren der Schalldruck, der über der Zeit aufgetragen ist, derart umgewandelt, daß danach die Frequenzen, aus denen sich das Geräusch zusammensetzt, als Linien im Spektrum erscheinen. Man betrachtet also ein zeitlich sich änderndes Ereignis aus einer anderen Sicht: der Sicht der Frequenzen.

Im Fall unseres Krimiauszuges könnte das wie in den Bildern 1 und 2 aussehen. Bild 1 zeigt, den Lautstärkepegel über

der Zeit dargestellt und Bild 2 läßt erkennen, daß drei ausgeprägte Frequenzen den Hauptanteil am Zustandekommen des in Bild 1 dargestellten Schalldruckgraphen haben. Man erkennt nun die individuellen Schlagzahlen der drei Steinmetze. Alles klar? Übrigens: Der Täter sitzt!

rk

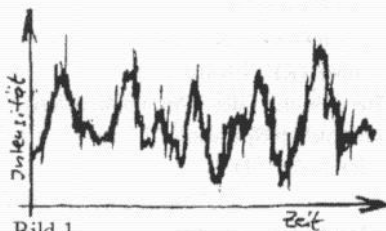


Bild 1

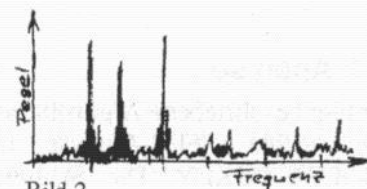


Bild 2

Stichworte

volksFORTH PC
Sinus
Wurzel
Bit-Reverse
FFT

Weitere Beispiele und den vollständigen Forth-Code für eine FFT finden Sie übrigens auch in DrDobbs' Toolbox Of Forth, Volume II (1987): Joe Barnhart, Forth and the Fast Fourier Transform.

Quadratwurzel-Berechnung

Die Quadratwurzel eines 32-Bit-Wertes wird ähnlich der Methode berechnet, wie man es vor vielen Jahren einmal in der Schule erlernt hat. Dadurch, daß im Binärsystem die Multiplikation zum Shiften um ein oder zwei Stellen wird, vereinfacht sich die Angelegenheit. Die gleiche Berechnungsart wird auch oft direkt in Assembler und meist nur mit den internen Register (falls groß genug) durchgeführt.

Im einzelnen läuft die Berechnung wie folgt ab:

- Summe und Wurzelwert auf 0 setzen
- Wiederholungsschleife (16 mal)
 - 2 Bits nach links (*4) von Quadrat in Summe shiften
 - Wert = Wert * 2 + 1
 - Test, ob Summe kleiner als die Wurzel
 - ja: Wert = Wert - 1
 - nein: Summe = Summe - Wert;
 - Wert = Wert + 1
- Summe und Quadrat vom Stack entfernen;
Ergebnis = Ergebnis/2

Bitumkehr

Bei der FFT-Analyse werden immer zwei komplexe Werte miteinander multipliziert. Um die Position der Werte zu bestimmen, wird auch eine sogenannte Bitumkehr benötigt.

Je nach gewählter Punkteanzahl (im Programm 256) werden die zur Bezifferung der Position notwendigen Bits (0.255 benötigen 8 Bits) in ihrer Reihenfolge umgedreht. Aus dem Binärwert %1110010 (=226) wird dann %0100111 (=71).

Durch Umsetzung dieser Routine in Maschinensprache läßt sich viel Zeit sparen, weil es schon im Beispiel über tausend mal verwendet wird.

FFT-Analyse

Der hier beschriebene Algorithmus ist eine echte FFT-Analyse nach COOLEY-TUKEY. Die Skalierung wurde so gewählt, daß immer der Scheitelwert der Sinus-/Cosinusschwingung und nicht der Effektivwert zurückgeliefert wird.

Der Kern des FFT ist eine komplexe Multiplikation zweier Werte, dessen Position durch den Wert des Schleifenzählers und die Bitumkehr abhängiger Werte bestimmt wird. Bei der

komplexen Multiplikation werden immer nur die obersten 16 Bit (gerundet) verwendet. Das entstehende 32-Bit-Ergebnis wird dann mit den gespeicherten 32-Bit-Wert verrechnet. Durch Umstellung dieser Routine auf volle 64 Bit-Zwischenwerte kann noch etwas Genauigkeit gewonnen werden.

Das Signal, das als 16-Bit-Wert (vorzeichenlos auf 32 Bit erweitert) im Real-Datenfeld vorliegt, wird der FFT-Analyse unterzogen und danach als Liste der Scheitelwerte zurückgeliefert. Die Adressierung von Signalposition und Amplitudenwert erfolgt mit dem Befehl RE (N . Dieser erwartet auf dem Stack die Nummer der Position (0..255) und liefert dessen Adresse zurück. Es muß immer ein 32-Bit-Wert gelesen und geschrieben werden. Nach der FFT-Analyse ist in 0 RE (N der Gleichspannungsanteil abgelegt.

Ablauf der FFT-Analyse im einzelnen:

- Vorbereitung des Datenspeichers
 - Imaginärteil löschen
 - Realteil mit 32768 multiplizieren
- Eigentliche FFT-Analyse
 - Dreifach ineinandergefügte Schleifen mit Kern KMU
 - Korrektur der Reihenfolge des Spektrums (REVERSEL)
- Berechnung der Amplitudenwerte aus komplexen Spektrum
 - Realteil = SQRT
(Realteil2 * Imaginärteil2)
 - Imaginärteil löschen
 - DC-Anteil korrigieren

Der Algorithmus selbst ist schon in vielen Büchern erwähnt und als Beispiel in vielen Programmiersprachen angegeben worden.

Beispiel FFTTEST:

Mit dem Befehl FFTTEST soll eine Schwingung der FFT-Analyse unterzogen und die ersten 10 Amplituden ausgegeben werden. Im Beispiel handelt es sich um ein Signal, das aus DC-Anteil und der 2ten, 5ten und 6ten Oberschwingung mit unterschiedlichen Amplituden zusammengesetzt ist. Es muß nur darauf geachtet werden, daß es kein Werteüberlauf in der 16-Bit-Darstellung gibt.

Übrigens können auch 32-Bit-Werte an die FFT-Analyse weitergereicht werden, wenn die Vorbereitung in FFT entsprechend abgeändert wird.

Die auch im Beispiel schon ersichtlichen Fehler entstehen hauptsächlich durch Rundungsfehler vor allem in den Sinus/Cosinus-Werten. Man kann aber damit leben, wenn das Ausgangssignal möglichst groß sind. Da bei mir die FFT-Analyse meist mit linksbündigen 12-Bit-Werten rechnet, stören die 2-3 Digit Ungenauigkeit kaum.

Literatur:

E. Oran Brigham FFT
- Schnelle Fourier-Transformation
(Oldenbourg-Verlag)

volksFORTH Spezialitäten:

```
&.... Zahl im Decimalformat ( &16384 )
RE bei FOR...RE...NEXT genutzt als
Index wie
bei DO...I...LOOP
I die folgende Definition ist HEADERlos
RESTRICT COMPILE only ( nicht im Direkt-Modus
akg
```

Forth und Buch

Hab ich richtig gelesen? 150.000 Neuerscheinungen auf der Frankfurter Buchmesse und kein Forth-Buch? Traurig, traurig! Was gibt es denn überhaupt noch auf dem deutschen Markt? Nun ja, immerhin bietet der Carl Hanser Verlag München nachwievor die beiden Leo-Brodie-Bücher (In Forth denken, 54,-DM und Programmieren in Forth, 52,-DM) an. Es wäre schön, wenn der Verlag sich entschließen könnte, Programmieren in Forth in der 2. Auflage heraus zu bringen. Mir liegt noch ein Verlagsverzeichnis 1992 des Verlag Technik GmbH vor, in dem unter der Nummer 226 das Vack-Buch Programmieren mit FORTH (1. Auflage 1991, 264 Seiten, 85 Bilder, 17 Tafeln, 38 Programme, Paperback, 42,-DM, ISBN 3-341-00518-8) angeboten wird. Die Luft wird ganz schön dünn, seitdem auch der Franzis-Verlag keine Forth-Bücher mehr führt. Da aber in zahlreiche Bibliotheken Forth-Bücher stehen, kann, bzw. muß über die Fernleihe auf Wunsch von jeder Stadt- und Land-Bücherei ein Forth-Buch beschafft werden. Die Nachfrage regelt bekanntlich den Markt. Regeln wir doch ein wenig! rk

Listing zu: Fourieranalyse in Forth (K. Kohl)

Screen # 0

```
\\ Sinusfunktion und FFT
System:      PC-volksFORTH 3.81.41
Programmteile:
```

1. Definition von FOR .. NEXT (Screen 2)
2. SIN, COS, SQRT, Bitumkehr (in der FFT-Analyse)
3. FFT-Programm (Screen 3-11)

Screen # 1

```
( LOADSCREEN ) ( 23.05.89/KK )
&02 LOAD ( Anpassung an volksFORTH )
&03 &10 THRU ( FFT-Analyse )
&11 LOAD ( Testroutine )
Screen # 2
( Definition des FOR..NEXT , ?NEGATE, D2/ ... ) ( 23.05.89/KK )
: FOR ( n - ; - sys ) IMMEDIATE RESTRICT
  COMPILER >R HERE $2468 ;
: (NEXT ( - ; R: n - n-1 | )
  >R > ?DUP IF 1- >R @ ELSE 2+ THEN >R ; RESTRICT
: NEXT ( - ; sys - ) IMMEDIATE RESTRICT
: ?NEGATE ( n f - n | -n ) ( -n, wenn f<0 )
  0< IF NEGATE THEN ;
: U2/ ( u - u/2 ) 2/ $7FFF AND ;
: D2* ( d - d*2 )
  OVER 2* -ROT 2* SWAP 0< - ; ( Bit 15 übertragen )
: D2/ ( d - D/2 )
  UNDER 2/ SWAP U2/ ROT 1 AND IF $8000 OR THEN SWAP ;
: D- ( d1 d2 - d1-d2 ) DNEGATE D+ ;
```

Screen # 3

```
( Variablen, Konstanten u. Felder ) DECIMAL ( 23.05.89/KK )
2 CONSTANT A/MW ( Adressen/16-Bit-Wert )
4 CONSTANT A/DW ( Adressen/32-Bit-Wert )
8 CONSTANT LD P ( 2^8 Punkte )
256 CONSTANT POINTS ( sind 256 )

VARIABLE LZ ( Endwert des aktuellen Exponenten )
VARIABLE NZ ( Endwert der aktuellen Meßwertnummer )
VARIABLE ZX ( 1. Meßwertnummer )
VARIABLE ZY ( 2. Meßwertnummer )

CREATE RE POINTS A/DW * ALLOT ( Realteil )
: RE(N ( n - addr ) A/DW * RE + ;
CREATE IM POINTS A/DW * ALLOT ( Imaginärteil )
: IM(N ( n - addr ) A/DW * IM + ;
```

Screen # 4

```
( SIN und COS ) ( 23.05.89/KK )
: (SP ( y - sign $16383 0 | sign y -1 )
  DUP ABS DUP $16384 =
  IF DROP $16383 0
  ELSE $16384 DUP ROT - ABS - -1
  THEN ;
: SIN ( 0..16384..32767..49152 - 0..32767..0..-32767 )
  (SP IF 2* 2* $25728 OVER DUP UM* NIP
  $10518 OVER $1173 UM* NIP -
  UM* NIP - UM* NIP
  THEN SWAP ?NEGATE ;
: COS ( 0..16384..32767..49152 - 32767..0..-32767..0 )
  $16384 + SIN ;
\ z =  $\frac{y^4}{65536} * ( 25728 - \frac{y^2 * 16}{65536^2} * ( 10518 - 1173 * \frac{y^2 * 16}{65536^2} ) )$ 
```

Screen # 5

```
( SQRT ) DECIMAL ( 23.05.89/KK )
( Quadratwurzel einer 32-Bit-Integerzahl )
( Zulässiger Bereich: $3FFF.0001 -> 0..32767 ( 15 Bit ) )
: SQRT ( ud - u )
  0 0 ( Summe und Wurzel )
  15 ( 16 mal )
  FOR >R 2* OVER 0< - 2* OVER 2* 0< - ( 2 Bits in S )
  >R D2* D2* R> R> ( Quadrat*4 )
  2* 1+ ( Wert*2+1 )
  2DUP U< ( Summe kleiner Wurzel ? )
  IF 1- ( dann nur Wert*2 )
  ELSE DUP >R - R> 1+ ( oder Summe=Differenz )
  THEN
  NEXT
  NIP NIP NIP U2/ ;
```

Screen # 6

```
( BITREV ) DECIMAL ( 23.05.89/KK )
: BITREV ( a - a2 ) ( Bitumkehr )
  0 SWAP LD(P 1-
```

```
FOR 2* SWAP OVER POINTS AND OR 2/ SWAP NEXT
DROP ;
: REVERSEL ( - )
  POINTS 1-
  FOR R@ BITREV DUP R@ U<
  IF R@ OVER >R >R R@ RE(N 2@ R> IM(N 2@
  R@ IM(N >R R@ 2@ 2SWAP R> 2! 2SWAP
  R> RE(N >R R@ 2@ 2SWAP R> 2!
  R@ RE(N 2! R@ IM(N 2!
  THEN DROP
  NEXT ;
```

Screen # 7

```
( weitere Routinen ) ( 23.05.89/KK )
\ Runden eines 32-Bit-Wertes auf 16 Bit
: ROUND ( d - n )
  DUP $32767 XOR
  IF OVER 0< - THEN
  NIP ;
```

Screen # 8

```
( FFT-Hauptroutine ) DECIMAL ( 23.05.89/KK )
: KMU ( - )
  ZX @ LZ @ ?DUP IF 1- FOR 2/ NEXT THEN ( ZX/2^LZ )
  BITREV [ 15 LD(P - ] LITERAL FOR 2* NEXT ( -> P )
  DUP SIN SWAP COS ( Sinus u. Cosinus )
  2DUP ZY @ RE(N 2@ ROUND M*
  ROT ZY @ IM(N 2@ ROUND M* D+ D2* ( S )
  2SWAP ZY @ IM(N 2@ ROUND M*
  ROT ZY @ RE(N 2@ ROUND M* D- D2* ( T )
  ZX @ IM(N 2@ D2/ 2OVER D- ZY @ IM(N 2! ( IM(Y )
  ZX @ IM(N 2@ D2/ D+ ZY @ IM(N 2! ( IM(X )
  ZX @ RE(N 2@ D2/ 2OVER D- ZY @ RE(N 2! ( RE(Y )
  ZX @ RE(N 2@ D2/ D+ ZY @ RE(N 2! ; ( RE(X )
```

Screen # 9

```
( FFT-Hauptroutine ) DECIMAL ( 23.05.89/KK )
: KFFT ( - )
  POINTS NZ ! LD(P LZ ! LD(P 1-
  FOR 0 ZX ! NZ @ 2/ NZ ! LZ @ 1- LZ !
  BEGIN NZ @ 1-
  FOR ZX @ NZ @ + ZY ! ( ZY=ZX+NZ )
  KMU ( Berechnung )
  ZX @ 1+ ZX ! ( ZX=ZX+1 )
  NEXT ZY @ 1+ DUP ZX ! POINTS U< 0=
  UNTIL
  NEXT
  REVERSEL ;
```

Screen # 10

```
( FFT ) ( 23.05.89/KK )
\ Signal im RE(N ( 16-Bit-Werte, wird auf 32 Bit erweitert )
\ wird in das Spektrum umgewandelt und zurückgeschrieben
: IMT ( - )
  IM POINTS A/DW * ERASE ( Imaginärteil löschen )
  POINTS 1- ( Realteil * 32768 )
  FOR R@ RE(N 2@ SWAP D2/ R@ RE(N 2! NEXT
  KFFT
  POINTS 2/ ( Amplitude aus Komplexwert )
  FOR R@ RE(N 2@ D2/ ROUND DUP M*
  R@ IM(N 2@ D2/ ROUND DUP M* D+ SQRT 2* 2* 2*
  0 R@ RE(N 2! 0. R@ IM(N 2! ( Wert in Realteil )
  NEXT POINTS 2/ DUP 1+ RE(N SWAP 1- A/DW * ERASE
  POINTS 2/ DUP 1+ IM(N SWAP 1- A/DW * ERASE
  0 RE(N DUP 2@ D2/ ROT 2! ( DC korrigieren )
  ;
```

Screen # 11

```
( Beispiel für FFT-Analyse ) ( 23.05.89/KK )
: FFTTEST ( - ) ( Oberwellen analysieren )
  $1.0000 POINTS UM/MOD NIP ( Grundschwingung )
  POINTS 1-
  FOR $1000. ( DC-Anteil )
  R@ 3 PICK * 2 * SIN 2 / 0 D+ ( + 8192*SIN[2] )
  R@ 3 PICK * 5 * COS 4 / 0 D+ ( + 4096*SIN[5] )
  R@ 3 PICK * 6 * SIN 8 / 0 D+ ( + 2048*COS[6] )
  R@ RE(N 2! ( speichern )
  NEXT DROP
  FFT ( FFT-Analyse )
  CR CR ." Analyse einer Schwingung: " CR
  9 FOR CR 9 R@ - DUP 3 U.R ." te Oberwelle: "
  RE(N 2@ DROP 6 U.R NEXT CR
  ;
```

LINKED ACTIONS und kein Ende

von Jörg Staben

LINKED ACTIONS in der VD 1990, 3 und 4 war zwar der Beitrag, für den ich - bildlich gesprochen - die meisten Schläge bekommen habe und ich hatte auch versprochen, so etwas nie wieder zu tun ... ABER ES LIESS MIR KEINE RUHE! Lesen Sie, mit welchem Erfolg.

der Reset- Taste. Kurze Überlegung: Wenn auf sich selbst oder ähnliche Kandidaten stößt, wird's wohl schiefgehen. Also wanderten BACK, TURN und W.EXECUTE in's Vokabular FORTH und schon lief die Sache. TURN sorgt dafür, daß mit WORDS alle Wörter eines Vokabulares ausgeführt werden, BACK stellt den friedlichen Zustand von WORDS wieder her.

Wahnsinn, ist das eine Sprache!



Da hatten wir uns hingesezt und hatten den Super-Listen-Handler mit einfügen, löschen, anhängen und vor allem - **ausführen** gebastelt. Schön, schön, es brachte zumindest ein Lob von Tom Zimmer ein. Es war auch wahnsinnig aufregend, Programmierung jenseits aller Borland-Compiler treiben zu können. Dummerweise wurde ich das Gefühl nicht los, alles hätte irgendwie einfacher gehen können. Denn FORTH an sich soll ja schon ganz gut mit Listen zurande kommen; es gibt da ein Wort, das neue Listen anlegt: VOCABULARY Ein Wort zum Einhängen neuer Elemente gibt's auch, es ist relativ bekannt: : Löschen kann man das letzte Element auch mit FORGET, nur das **Ausführen** einer Liste, das gibt's nicht. Wirklich nicht? Wenn ich mir so anschau, was WORDS so macht... Man kann offensichtlich mit solch einer Vokabular-Liste wohl etwas anstellen, vielleicht kann man sie auch ausführen.

Ein Blick in's F-PC zeigt endlich mal eine Vorwärtsreferenz an der richtigen Stelle: DEFER W.ID Ganz zart und zaghaft wird ein Wort gebaut, das WORT.ANZEIGEN gegen WORT.AUSFÜHREN austauscht: W.EXECUTE Schnell noch zwei Wörter zum Hin- und Herschalten gebastelt und ... der Rechner ruft nach

Listing zu: Linked Actions (Jörg Staben)

```

\\ Woerter in verketteten Listen - Linked Actions, Teil III
jrg 04.05.92

Ketzerische Frage:
Ist FORTH selbst schon alles, was man fuer eine Liste braucht?
{
only FORTH also definitions

anew linked_actions
}
***** JRG 04.05.92 21:02:14.29 *****
W.EXECUTE TURN und BACK
muessen vor dem Ausfuehren geschuetzt werden.
Deshalb im Vokabular FORTH definiert.

TURN schaltet das Ausfuehren von Wörtern ein,
WORDS führt dann alle Wörter im Vokabular aus;
BACK schaltet auf das normale Verhalten von
WORDS zurück.
*****
{
: w.execute ( nfa -- ?? )
name> execute
;

: turn ." TURNed to execute "
['] w.execute is w.id ;

: back ." BACK to show "
['] .id is w.id ;
}
***** JRG 04.05.92 21:03:10.97 *****
Dieses Vokabular wird ausgefuehrt!
*****
{
Vocabulary TEST
TEST also definitions

: zeigen ." Das war's!" ;

: Taste ." Taste: "
key emit
;

: W ." Welt " ;

: H ." Hallo " ;

: cr cr ;

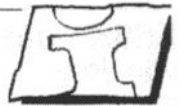
\s

```

Die Anwendung ist nicht ganz trivial. Sie testen diese Idee mit:
TEST words
turn TEST words
back TEST words
Wenn TURN aktiviert ist, darf in KEINEM Fall irgend ein Wort hinter WORDS stehen, sonst sanftes Aufhängen. Die Ausnahme der Regel:
turn TEST these words e
back
Happy back-turning akg



Stichworte
F-PC
LINKED ACTIONS
List execute



F68K - das erste Jahr

von Jörg Plewe

Auf der Forth-Tagung 1990 in Frankfurt wurde bedauernd festgestellt, daß es keine preiswerte, leistungsfähige und leicht verfügbare Forth-Implementation für Amiga-Computer gibt. Das daraufhin entwickelte Freeware-System F68K hilft dem ab und läuft zugleich auch auf jedem anderen Rechner mit M680x0 CPU. Dieser Beitrag gibt einen Überblick über den Stand der Entwicklung und über Zukunftsperspektiven.

Geschichtliches

Die Entwicklung des Systems F68K begann im Sommer 1990. Ich hatte auf der Tagung der Forth Gesellschaft 1990 in Frankfurt versehentlich zugesagt, eine lauffähige Forth-Implementation für Commodore Amiga bereitzustellen.

Bei dieser Gelegenheit und durch das Fehlen einer entsprechenden Entwicklungsmaschine faßte ich den Entschluß, vom speziellen Rechner zu abstrahieren und bei der Codeerzeugung lediglich die CPU, nicht aber den gesamten Rechner zu berücksichtigen. Später sollte dann auf einfache Weise die Anbindung an den die CPU umgebenden Rechner erfolgen können.

Eine der Hauptzielsetzungen war eine geradlinige und durchschaubare Struktur. Es sollte möglich gemacht werden, Forth Quelltext ohne weitere Beachtung von Systeminterna aus fremden Quellen übernehmen und lauffähig machen zu können. Solche Forderungen sind den Versuchen entsprungen, Quelltexte aus 'Object Oriented Forth' u.ä. Werken auf komplexen Systemen wie F-PC oder meinem FFORTH zum laufen zu bringen. Gehashte Vokabulare und raffinierte Methoden der Codeerzeugung, wo-

möglich mit vor/bei/nachgeschalteten Optimierern, ließen derlei Versuche gerne scheitern. Hinzu kam eine Weisheit aus dem Profibereich, daß sich der praktische Wert eines Systems gar nicht erhöht, wenn es 10% schneller wird. Wenn dabei die interne Komplexität zunimmt, wird man sogar einen Wertverlust hinnehmen müssen. F68K wurde unter diesem Eindruck auf der Basis von FFORTH entwickelt. Es ist aber nicht als Weiterentwicklung zu verstehen, da der mit FFORTH beschrittene Weg - ein möglichst effizientes System für einen speziellen Rechner - nicht weitergegangen, sondern in wesentlich Punkten umgekehrt wurde.

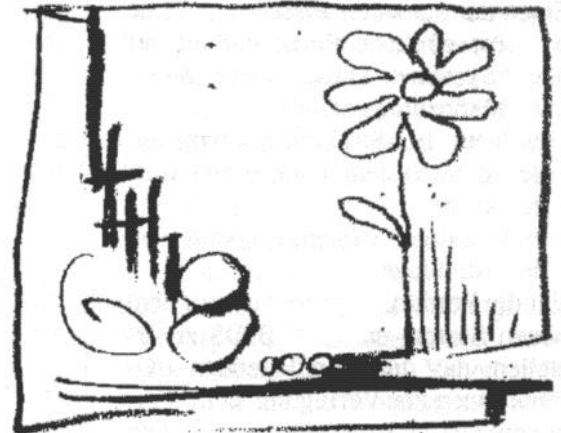
F68K hat nun sein erstes Jahr als voll funktionsfähiges und vor allen Dingen verfügbares System hinter sich gebracht.

Technisches

Es erforderte einigen Aufwand, aus den mehrfach (16) gefädeltten Vokabularen des FFORTH wieder schöne, einfache und lineare Strukturen für F68K zu erzeugen. F68K-Vokabulare sind wie in frühen Tagen schlichte, lineare Listen. Wer 'Object Oriented Forth' gelesen hat, kennt deren Wert. Kontrollstrukturen, in FFORTH noch jeweils in optimale Maschinenbefehle aufgelöst, benutzen in F68K nun wie-

der die BRANCH- und ?BRANCH-Primitiven. Variablen und Konstanten werden nach guter Väter Sitte ganz gewöhnlich mit CREATE und DOES erzeugt und nehmen dadurch einen gewissen Laufzeitoverhead in Kauf, der durch recht einfache Optimierungen im FFORTH vermieden wurde. Wohlgedemerkter, dies alles hat Programmieraufwand gekostet, nicht gespart! Der Gewinn: das Aussehen des Codes ist wieder vorhersagbar geworden und neue Kontrollstrukturen sind konsistent einzufügen. Für F68K wurde reumütig von Streamfiles zu Blöcken zurückgekehrt, da sich im direkten Vergleich stets die geblockte Überlegenheit herausgestellt hat.

Obwohl dies alles nach einem back-to-the-roots FIG-Forth von 1975 klingt, hält F68K auch für den nach Neuerungen rufenden Entwickler einiges parat. Es mag vielleicht schon aufgefallen sein, daß es für einige Rechner heute bereits C-Compiler



gibt. Diese erzeugen in der Regel mehr oder weniger guten 'native code'. Forth bzw. F68K sollte sich wenigstens in der Größenordnung mit solchen Entwicklungsmöglichkeiten messen können. Deshalb erzeugt auch F68K (mehr oder weniger) guten 'native code'. Hierbei wurde allerdings aus den oben genannten Gründen lediglich das Prinzip der 'native code' Generierung angewandt. Das Kompilat bleibt einfach und durchschaubar. Konzeptionell tendiert F68K in Richtung des kommenden ANSI-Standards. Einige ANSI-Pflichtübungen wie VALUES oder LOCALs turnt F68K bereits mit Leichtigkeit und durchweg guter Performance.

Stichworte

F68K
Portierung
Implementationen

Auch bei der Quelltextverwaltung geht F68K neue Wege. Wie bereits erwähnt, greift F68K wieder auf Blöcke zurück. Diese wurden allerdings auf 2kB erweitert. Dies hat neben dem verbesserten Komfort beim Programmieren auch weniger Schwierigkeiten beim Import von Streamfile-Quelltexten zur Folge, da diese i.d.R. das 64-Spaltenlimit überschreiten. Die F68K Blocks sind zudem in einem intern organisierten Filesystem namens FIFI (Forth Integriertes File Interface) untergebracht. Dieses stützt sich lediglich auf das Forthwort BLOCK und portiert sich damit ebenso einfach wie F68K selbst. Es hat sich gezeigt, daß FIFI-Files und gewöhnliche Streamfiles recht einfach ineinander zu überführen sind. Auch alte Blockfiles lassen sich vollautomatisch und verlustfrei importieren.

Portierung

Einer der stärksten Punkte im F68K ist seine einfache Portierbarkeit auf neue Systeme. Diese wurde durch zwei Maßnahmen erzielt:

- einfache I/O-Struktur, die von außen dem System vorgegeben werden kann,
- vollständige Unabhängigkeit von der Adreßlage.

Für die Portierung genügt es, auf dem neuen System eine Art BIOS zu erstellen, das die grundlegenden I/O-Funktionen zur Verfügung stellt. Ferner muß dafür Sorge getragen werden, daß das F68K Systemimage irgendwie in den Speicher gelangt. Wenn dann noch einige Adressen für Stacks und TIB erfunden werden können, sollte F68K schon laufen. Aufwendige Relozierrmechanismen, wie sie im FORTH noch notwendig gewesen waren und zum Teil bis auf Forth-Ebene durchschlugen (Stichwort: AVARIABLE), entfallen völlig.

Daß dies nicht nur in der Theorie funktioniert, konnte F68K schon mehrfach unter Beweis stellen.

Umfang

Durch die Mitarbeit einiger engagierter Entwickler konnte F68K einen an-

Wortliste des F68K

forth words	L	V	EDITOR
STRINGS	^F	^V	^C
^A	^W	^U	^Z
^L	^K	^P	^O
^	^B	^N	^?
^T	^Y	^Q	^R
^=	^H	ESC	^M
^J	^I	^X	^E
^D	^S	^	^G
FORTH-83	FORTH-83-VOC	/JSR	MAKE
INCLUDE	RESTORE_STREAM	SAVE_STREAM	VIEW
VIEWLIST	APPENDSTREAM	FILLSTREAM	COPY
COPYERR"	COPYBUF	TOSTREAM	FROMSTREAM
AT	NOWRAP	WRAP	NORMAL_VIDEO
REVERSE VIDEO	_DEL SOL	ERASE_LINE	RESTORE_CURSOR
SAVE_CURSOR	_CUTOFF	HIDE_CURSOR	CURON
SHOW_CURSOR	DEL SOP	DEL_LINE	INS_LINE
DEL_EOL	DEL_EOP	CUR_UP+INS	HOME
PAGE	CUR_LEFT	CUR_RIGHT	CUR_DOWN
CUR UP	ESC:	#ESC	ENDCASE
DEFAULT	ENDOF	OF	CASE
EDERROR	ERRORSTREAM	R#	LIST
LINE	SCR	C!+	W!+
!+	C@+	W@+	@+
TYPE16	DUMP	STOP?	FIRSTLINE
DUMLINE	?TERMINAL	.R	.S
DEPTH	THRU	PERFORM	[JSR_ONLY]
RDROP	2R	2R	U/
MOD	OFF	ON	TRUE
FALSE	CS	CLEARSTACK	?
BOUNDS	.LAST	ONLYFORTH	-TRAILING
[VOC']	VOC'	VOCS	ORDER
(ORDER	.VOC	>BODY	.NAME
NAME	PFA>CFA	(PFA>CFA	EMPTY
?EXIT	WITHIN	0<>	NIP
UNDER	RP!	RP@	FP!
FP@	RECURSIVE	LW	LB
WL	BL	6-	6+
8/	8*	8-	8+
4/	4*	4-	4+
-!	-ROLL	ROLL	PICK
-ROT	EVEN	#IFDEF	#IFNDEF
\NEEDS	?16CR	?CR	TAB
SPACES	BACKSPACES	BACKSPACE	BELL
COLUMNS	ROWS	(ABSCODE)	(ABS)
ABSCODE'	>ABS	;CODE	;ASS
OCTAL	MCODE	CODE	ASSEMBLER
XX	AFTER_LOAD:	EXPEL	INSERT
LESS	MORE	DELDIR	DEL
CD	PWD	(PWD	WHATDEVICE
USE	USESTREAM	MAKEDIR	MAKE
D3:	D2:	D1:	D0:
-->	DIRECT	STREAMS	SLOAD
SBUFFER	SBLOCK	FORMATTED?	STREAM:
DIR	COUNT0	MOUNT	UNMOUNT
CM	CHANGEMEDIUM	FORMAT	?BUSE
BTHRU	LOADSTREAM	BBUFFER	BLOAD
BBLOCK	SBLK	BLOCKSTREAM	BYE
FLUSH	EMPTY-BUFFERS	(CLEAR-BUFFER	CLEAR-BUFFER
SAVE-BUFFERS	SAVE-SYSTEM	WRITE-CODE	WRITE-HEADER
MAGIC	IS	CODE@	CODE!
CODE>DATA	STOP?	?TERMINAL	2CELLS+
MOD	-TRAILING	NIP	UNDER
2DUP	2DROP	-ROT	USER
UALL0T	EVEN	THRU	\
\	NULL	CONTEXT	LOAD
{LOAD	UPDATE	CHARS	CHAR+
CELLS	CELL+	2/	2*
2-	2+	1-	1+
ERASE	FILL	VOCABULARY	RESTRICT
IMMEDIATE	POSTPONE	"	ABORT"
ABORT	ERROR"	{	.(
."	."	FORGET	(FORGET
MAX	MIN	<=	>=
0<=	0>=	0<	0>
0=	WHILE	REPEAT	AGAIN

Fortsetzung der Wortliste des F68K

UNTIL	BEGIN	ELSE	BRANCH
THEN	IF	?BRANCH	LEAVE
+LOOP	?DO	LOOP	DO
<RESOLVE	<MARK	>RESOLVE	>MARK
UNLOOP	J	I	(+LOOP
(LOOP	(?DO	(DO	LOCAL
VALUE	TO	BL	CONSTANT
VARIABLE	DEFER	;CODE)	DOES>
CREATE	CMOVE>	CMOVE	2SWAP
2OVER	2DUP	2DROP	OVER
?DUP	ROT	SWAP	DROP
DUP	;	REVEAL	M:
:	HEADER:	DECIMAL	HEX
W!	C!	!	We
C@	@	NNUMBER?	CONVERT
COUNT	ACCUMULATE	DIGIT?	COLD
QUIT	ALIGN]	[
PROMPT	U.	#S	SIGN
HOLD	#>	<#	EVALUATE
SAVEAREA	PUSH	INTERPRET	PARSER
INTERPRETER	COMPILER	?STACK	[']
,	H'	UNKNOWN	NOTFOUND
NULLSTR?	FIND	(FIND	VOCSEARCH
NAME	CAPITALIZE	CAPITAL	[CHAR]
CHAR	WORD	SOURCE	SCAN
SKIP	QUERY	TIB	BLOCK
(BLOCK	BUFFER	(BUFFER	LASTBUF
LASTBLK	?CORE	(ABORT"	(ERROR"
(. ")	(")	FLITERAL	FLIT
FLOATLIT	LITERAL	LIT	SPACE
CR	R@	R>	R<
SP!	SP@	EXECUTE	EXIT
ALLOT	ABS	NEGATE	NOT
XOR	OR	AND	*/
*/MOD	/	/MOD	U/MOD
*	-	+	+
F,	C,	W,	JRSRB,
CODEW,	CODE,	COM,	JSR,
,	TYPE	EXPECT	READSYS
WRITESYS	R/W	KEY?	EMIT
KEY	LOADERREADSYS	LOADERWRITESYS	LOADERR/W
LOADERKEY?	LOADEREMIT	LOADERKEY	NUMBER?
HERE	PAD	OUT	UDP
CAPS	USERBUFS	PREV	ROOTBLK
BLK	FWIDTH	WARNING	IS MACRO
MACRO	(FLITERAL)	(LITERAL)	(TYPE)
(EXPECT)	^WRITESYS	^READSYS	^R/W
^KEY?	^EMIT	^KEY	(WRITESYS)
(READSYS)	(R/W)	(KEY?)	(EMIT)
(KEY)	(ERROR)	LAST	VOCPA
VOC-LINK	CURRENT	SPAN	IN
#TIB	TIB	DP	CP
HLD	DPL	BASE	(NUMBER?)
STATE	FO	SO	RO
NEXTUSER	VER	NOOP	(END OPT)
(FRONT_OPT)	FENCE	WRITESYSES	READSYSES
R/WS	KEY?S	EMITS	KEYS
ROOTTABLE	FORTHPARAS	DATABOT	DATATOP
SYSBOT	SYSTOP	(COLD)	BYE
OSTYPE	OSEXPECT	PAUSE	

Hinzu kommen weitere Worte aus den anderen Vokabularen:

vocs
EDITOR STRINGS FORTH-83-VOC ASSEMBLER BLOCKSTREAM FORTH ONLY

Abb. 1

sehnlichen Umfang erreichen. F68K will dabei nicht Systemen wie F-PC nacheifern. Es kommt vielmehr darauf an, Wichtiges für die Entwicklung mitzuliefern, anstatt alle mögli-

chen Entwicklungen durch passende Bibliotheken vorwegzunehmen. Neben den im wesentlichen von mir erstellten Erweiterungen des Kerns, die jedes Forth-System enthalten muß

(WORDS, DUMP, .S ...), sind einige wichtige Aufsätze hinzugekommen. Mitgeliefert werden (Stand April '92):

- **FIFI**
Jörg Plewe
- **Fullscreen Editor/Tools**
Markus Redeker
- **Assembler**
Laxen&Perry, Hartwig Lüdemann
- **Disassembler**
Reinhard Schamagl
- **Floatingpoint**
mc, Thomas Krutmann,
Jörg Plewe

Hinzu kommen weitere nützliche (z.B. VALUE- und LOCAL-Generator für verschiedene Datentypen) oder einfach nur schöne Tools (z.B. UCS - Universelle Kontrollstrukturen).

Außerdem haben sich Ergänzungen eingefunden, die sich auf spezielle Rechner/Betriebssysteme beziehen. So hat *Markus Redeker* einige Atari-TOS-Aufrufe eingebunden, und *Rainer Saric* hat versprochen, sein GEM-Paket, das schon in ForthMacs, FFORTH und BigForth seinen Dienst tut, auch auf F68K zu portieren. Für den Sinclair QL existiert von *Dirk Kutscher* die bisher vollständigste Anbindung, die ich allerdings nicht beschreiben kann, da ich diesen Rechner überhaupt nicht kenne.

Zur Dokumentation des Umfangs des Standard-Entwicklungssystems (Kernel + Streams aus dem Verzeichnis /SYSTEM) ist die Wortliste (Abb. 1) wiedergegeben.

Der Inhalt des FIFI-Systems gliedert sich im Standard-Lieferumfang, entsprechend Abb. 2.

Nach dem Laden der wichtigsten Streams aus dem Verzeichnis /UTILITY ist die Anzahl der Vokabulare bereits heftig angestiegen. Ihre Aufzählung erübrigt eigentlich die genaue Darstellung dessen, was da eigentlich geladen wurde:

UCS
LOCALS
VALUES
FLOAT
ON-LINE-DOCUMENTATION
EDITOR
STRINGS
FORTH-83-VOC

Standart-Lieferumfang des F68K

Gerät D0:		
Free		49 blocks
SYSTEM		>dir<
	..	>dir<
	STREAMUTILS	5 blocks
	LOADME	1 blocks
	SAVE-UTILITY	1 blocks
	LINE-EDITOR	4 blocks
	L&P-ASSEMBLER	16 blocks
	VT52-TERMINAL	2 blocks
	COPYBLOCKS	1 blocks
	CASE	1 blocks
	F68K-ASS	1 blocks
	KERNEL-EXTENSIONS	27 blocks
	COMPATIBILITY	2 blocks
	TERMINAL	1 blocks
	F-EDIT	>dir<
	..	>dir<
	F-EDIT	18 blocks
	STRINGS	1 blocks
	CONTROLKEYS	1 blocks
	STANDARD-EDIT	3 blocks
	ATARI-EDIT	3 blocks
UTILITY		>dir<
	..	>dir<
	FLOAT	>dir<
	..	>dir<
	LOADFLOAT	1 blocks
	FLOATDUMP	10 blocks
	FLOAT-BINDING	16 blocks
	ANSI	>dir<
	..	>dir<
	ANS-LOCALS	5 blocks
	ANS-VALUES	4 blocks
	ANS-LOCALS_(TO)	5 blocks
	FACTORIAL	2 blocks
	IMPORT-EXPORT	2 blocks
	BADEN'S-:DOES>	1 blocks
	PAYSAN'S-LIST>	1 blocks
	LOADME	1 blocks
	ON-LINE-DOC	3 blocks
	STRINGS	1 blocks
	AUTO	1 blocks
	UCS	8 blocks
	DISASSEMBLER	13 blocks
SINCLAIR-QL		>dir<
	..	>dir<
	>EMITS	1 blocks
	QDOS-CALLS	8 blocks
	PARAMETER	2 blocks
	LOAD_QL_WORDS	1 blocks
	TRAP1-CALLS	3 blocks
	TRAP2-CALLS	3 blocks
	TRAP3-CALLS	6 blocks
	ANWENDUNGEN	>dir<
	..	>dir<
	SIEVE_BENCH	3 blocks
	JOB_CONTROL	8 blocks
	EXAMPLES	1 blocks
	TERMINAL	2 blocks
	TALK	7 blocks
	FILESERVER	1 blocks
ATARI		>dir<
	..	>dir<
	OS-CALLS	2 blocks
AUTOINCLUDE		
Gerät D1:	unformatiert, enthält Definition des FIFI	

Abb. 2

ASSEMBLER
BLOCKSTREAM
FORTH
ONLY

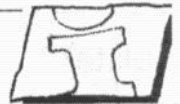
Seit Kurzem existiert ein weiteres, großes Gerät D2: mit 1000 Blöcken. Auf diesem Gerät sollen weitere Sourcen, die nicht unmittelbar mit dem System zusammenhängen, sondern mehr Anwendungs- oder Democharakter haben, untergebracht werden. Zum Zeitpunkt der Niederlegung dieses Artikels enthält es allerdings erst die Sourcen zu dem Spiel CORE-WARS, die von einer 'User Contribution Disk' der FIG portiert wurden. Die Auslieferung dieses Gerätes ist etwas problematisch, da es auf der Festplatte des Wirtsrechners mit 2MB zu Buche schlägt und somit nicht mehr einfach auf eine Diskette zu bringen ist. Je nach persönlichem Bedarf wird es in geARCTer oder vollständig ins DOS exportierter Form erhältlich sein. Spezialwünsche können natürlich ebenfalls geäußert werden.

Vorhandene Implementationen

Für verschiedene 68000-Rechner existieren bereits fertige Ladeprogramme, die sich mehr oder weniger gut in ihre Umgebung einfügen. Sie basieren zur Zeit alle im wesentlichen auf einem von mir in C verfaßten Lader.

- **Atari ST**
Jörg Plewe
- **Commodore Amiga**
Wolfgang Schemmert
- **OS9**
Dr. Thomas Beierlein
- **Sinclair QL**
Dirk Kutscher
- **GEM in Vorbereitung**
(Jörg Plewe)

Hinzu mag da noch die ein oder andere Installation auf Einplatinencomputern kommen. Da das Echo von F68K-Anwendern in der Regel aber äußerst bescheiden ist, ist mir hierüber nicht allzuviel bekannt (genaugenommen weiß ich auch nicht, ob die Anwender wirklich Anwender sind, oder ob sie nur über zwei weitere Leerdisketten verfügen).



Dokumentation

F68K wird derzeit mit etwas weniger als 150 kB (touristen-) englischer Dokumentation in einem ASCII-File ausgeliefert. Dieses 'Technical Reference Manual' enthält neben der Erläuterung der wichtigsten Systemstrukturen eine genaue Anleitung zur Portierung und die Dokumentation der bereits vorhandenen Implementationen.

Diesen Zustand empfinde ich als unbefriedigend, da die Doku beim Übergang von meinem Rechner zum Anwender erheblich an Struktur verliert (bei mir befindet sich z.B. das Glossar in einer Datenbank). Die Weitergabe auf Diskette verlangt aber ein ÜBERALL maschinenlesbares Format. Deshalb führt an ASCII wohl kein Weg vorbei. Grafiken oder ein wirklich guter Ausdruck bleiben damit aber mir selbst vorbehalten.

Um diesen Mißstand beiseite zu schaffen, prüfe ich derzeit die Möglichkeit, die Dokumentation im Format des Satzsystems TeX/LaTeX auszuliefern. Dies ist allerdings noch nicht spruchreif, so daß ich diese Möglichkeit hier nur erwähnen möchte. Kommentare erwünscht!!

Zukunft

Programmiersysteme entwickeln sich weiter. Vor dieser Tatsache wird auch Forth die Augen nicht mehr lange verschließen können. Es nützt nicht viel, die ohnehin schon ungemein weiten Grenzen der SPRACHE Forth noch weiter hinauszuschieben. Auch die UMGEBUNG, also die Seite eines Systems, mit der der Entwickler seinen Tag verbringt, bedarf der Aufmerksamkeit, will Forth nicht im Turbo-Dschungel untergehen. Selbst in der Softwareentwicklung für eingebettete Systeme, einst Domäne von Forth, machen sich GEM und WINDOWS-gestützte Systeme breit. Anwender lieben das und werden über kurz oder lang die erweiterten Möglichkeiten der Sprache Forth den Angenehmlichkeiten 'moderner' Systeme opfern. Denn intuitive Oberflächen haben mächtig an Funktionalität gewonnen und einige Umständlich-

keiten abgelegt. Wer heute noch nur das für 'professionell' hält, was hart macht, wird irgendwann von einer Welle neuer Applikationen und Entwicklungsumgebungen überrollt werden. Daß es auch anders geht, beweisen Systeme wie F-PC, die mit integrierter Umgebung und ausgeklügeltem Hilfe- und Quelltextsystem auch den verwöhnten PC-Markt in für Forth untypischer Weise erreichen können.

Auch auf 'Spielzeugcomputern', wie den Rechnern der Atari-Reihe, mit denen ich mich plage, schalten fast alle Forth-Implementationen den zum Standard gewachsenen und für die Zukunft unverzichtbaren GEM-Komfort ab und präsentieren sich wie eh und jeh:

```
ok
input x
INPUT unknown!
ok
exit
EXIT compile only !
ok
quit
ok
system
SYSTEM unknown !
ok
```

F68K sieht in seiner jetzigen Form kein bißchen besser aus. Die Konkurrenz wird dieses Stadium wohl nur noch 'die frühen Jahre' nennen, in denen so etwas sogar noch verkäuflich war. Heute aber hat man Schwierigkeiten, solche Systeme noch in einem PD-Versand unterzubringen. Und die sind bestimmt nicht anspruchsvoll. Lediglich ein Low-Cost-System aus dem Verlag M&T hat es fertiggebracht, diese Schwelle zu überschreiten und ein Forth-System in eine GEM-Oberfläche einzubinden. Auch andere wichtige Reizworte wie z.B. Multitasking erledigte es mit links. Leider hat man es dabei aber veräußert, andere wichtige Forth-Eigenschaften beizufügen, so daß sich dieses System (leider) nicht durchsetzen konnte. Die meisten anderen Systeme, die auf dem Markt erhältlich sind, namentlich BigForth, FORTH und ForthMacs (in alphabetischer Reihenfolge), verfügen zwar über 'ein bißchen GEM', doch das wird wohl für

die Zukunft nicht ausreichen. Sie ermöglichen die Erstellung komfortabler Anwendungen. Das Leben des Entwicklers aber bleibt hart. Vielleicht sollte es umgekehrt sein?? Wenn man als Aktivist der FG Forth nicht nur benutzen, sondern auch nach draußen tragen will, muß man in der Lage sein, eine breitere Öffentlichkeit anzusprechen. Und diese ist entsprechend verwöhnt. Jeder Hobbyprogrammierer erwartet von JEDEM Programm, daß es mit GEM-Multitasking-erweiterungen verträglich ist und daß er seine lieb gewordenen Helferlein jederzeit im Zugriff hat. Dieser Anspruch ist legitim, setzt aber den Einsatz von GEM immer voraus. Für F68K habe ich daher die Konsequenz gezogen und mit der Entwicklung eines GEM-Laders begonnen. Bemerkenswerterweise liegt damit erstmalig die Komplexität eines F68K-Laders in der gleichen Größenordnung wie das System selbst.

Noch einmal zur Erinnerung:

- F68K ist für Mitglieder der FG gegen Kostenerstattung zu haben. Vertrieb: Johannes Teich (siehe letzte Seite)
- Nichtmitglieder kostet es soviel wie ein FG-Jahresbeitrag + Kostenerstattung.

Nachtragend . . .

. . . ist Jörg Staben zum Glück nicht! Er hat mir verziehen, daß ich sein Konterfei im Direktorial des Heftes 8/2 verunstaltet habe. Deshalb liefere ich jetzt auch den folgenden Quelltext nach, den er im Light-Show-Artikel (VD 8/2, S. 12) erwähnte. rk

```
: .logics
  dual
    2 0 DO
      2 0 DO
        cr
        i j . . . " AND: "
        i j and . 2spaces
        i j . . . " OR: "
        i j or . 2 spaces
        i j . . . " XOR: "
        i j xor . 2 spaces
      LOOP
    LOOP
;
```

Ein klein, fein Helferlein

von Jörg Staben

Den Screen printen aus F-PC (nach einer Anregung von W. Engler, München)

PRTSCR ist ein wunderschönes Beispiel dafür, wie kurz und knapp ein nützliches Programm sein kann. Zwar ist auf einem PC eine Bildschirm-Hardcopy jederzeit interaktiv auf DRUCK oder SHIFT- PRTSCR hin möglich, aber oftmals möchte man diese Funktion aus einem Programm heraus aufrufen.

Stichworte

Bildschirm Ausdruck
Interrupt
bedingtes Compilieren
F-PC 3.56
TCOM 8086
IBM-PC

Zugleich bewirbt sich PRTSCR um den Titel des kleinsten nützlichen Programmes, das von TCOM immerhin auf nur 36 Byte übersetzt wird - ein Wert, der z.B. für ein TurboPASCAL (noch) unerreichbar ist. Für ein TPASCAL liegt die minimale Codegröße bei 1,5 KByte, die MS-C Compiler verbrauchen sogar immer direkt 2 KByte. Dennoch ermöglicht die Kombination F-PC/TCOM die unschätzbare wertvolle Möglichkeit des interaktiven Ausprobierens.

Sollten Sie weitere kleine, knuffelige Programme in Ihrer Schublade haben, ist jetzt die Gelegenheit, die Schleusen zu öffnen und die Redaktion der VD damit zu überfluten.

P.S.:

Nicht vergessen: Im TCOM kann das F-PC mit in die Entwicklungsumgebung eingebunden werden! Dies leistet ein Eintrag in der Datei ZZ.CFG z.B. an der Position SHIFT-F9:

```
c:\fpc\f %1 ok
```

Durch das OK wird der Quelltext direkt kompiliert. Verlassen Sie das F-PC mit BYE, so befinden Sie sich sofort in der ZZ-Entwicklungsumgebung. \FPC und \TCOM sorgen übrigens dafür, daß jeder Compiler nur die Zeile "sieht", die seinen Namen trägt.

```

\ Hardcopy aus dem Programm heraus
.options /opt /oinit
)
**** jrg 05/01/92 22:14 *****
PRTSCR
      löst eine Harcopy aus.
      Idee: Wolfgang Engler, MÜNCHEN
*****
{
Code prtscr ( -- )
      int 5
\ fpc      next
\ tcom    ret
      end-code

\s

```

F68K-Neuigkeiten in letzter Minute

von Jörg Plewe

Ergänzungen zum Artikel auf den Seiten 17-21

Neuigkeiten gibt es diesmal nur für Besitzer der Atari-Linie. Wie nicht anders zu erwarten, verträgt sich F68K einwandfrei mit der Multitasking-erweiterung MiNT (MiNT is Not TOS). MiNT ist PD und erlaubt den Betrieb von mehreren Programmen gleichzeitig. MiNT ist zugleich die Basis für Ataris neues Betriebssystem MultiTos, so daß sich wohl früher oder später alle Programme damit anfreunden müssen.

Programme ohne eigene GEM-Oberfläche können dabei mit Hilfe des Accessories BACKGROUND.ACC in einem GEM-Fenster ablaufen. Leider kann man z.Zt. noch nicht allzu viele Programme unter MiNT multitasken lassen. F68K und ForST aus der FIG

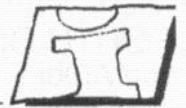
User Contribution sind die einzigen mir bekannten Forth-Systeme, die problemlos mit MiNT bzw. BACKGROUND.ACC funktionieren. Alle anderen gehen den Bach 'runter (bigForth 1.12, Forthmacs 1.1; die übrigen - 32Forth, FForth, volksForth und 4xForth - habe ich gar nicht erst probiert, da sie entweder mit dem TT oder dem Großbildschirm hadern oder gar nicht mehr zur Debatte stehen. Dies gilt für Atari TT/Großbildschirm und vorbehaltlich neuerer Systemversionen; bei bigForth habe ich da einige Hoffnung.)

So kann man jetzt bis zu sieben (= max. Fensterzahl) F68K auf dem Desktop oder in einer laufenden Applikation ablegen. Hierbei ist aller-

dings doch Vorsicht geboten, da das F68K-Filesystem FIFI von der Anwesenheit seiner Brüder nichts ahnt und so durch unvorsichtigen, schreibenden Zugriff aus verschiedenen Systemen Schaden nehmen kann.

Meistens kommt man aber mit einem einzigen System schon ganz gut hin. Die gleichzeitige Arbeit mit F68K im Fenster und dem Desktop oder einer GEM-Anwendung verleiht dem System eine neue Qualität und es entsteht der Eindruck, als sei F68K für den Atari geschrieben worden. Wer noch weiter gehen will, kann F68K auch unter der für MiNT verfügbaren grafischen Oberfläche MGR starten. Auch hier natürlich keinerlei Probleme.

Mit dem Kommando half vt52 f68k startet F68K in einem eigenen Fenster. Dies kann man treiben, bis der Speicher (und der Bildschirm) voll ist. Alles funktioniert wie immer.



FORTH- RISC-Prozessor FRP 1600

von Arndt Klingelberg

Wesentliche Elemente der virtuellen Forth-Maschine in Silizium, Schnelligkeit in Software-Entwurf und Ausführung für Echtzeit verbunden mit Interruptbehandlung, MultiTasking, Bild- und Signalverarbeitung.

Bei Forth-Prozessoren ergibt sich die Besonderheit, daß der Maschinen-/AssemblerBefehlssatz schon Jahrzehnte getestet und auch in Normenausschüssen beraten wurde. Das ist wohl einzigartig. Für den FRP 1600 gilt zudem, daß in ihn die Erfahrungen mit bisherigen Chips wie Harris RTX-2000 eingeflossen sind, und vieles, was dort noch etwas 'klemmte', hier optimiert wurde. Daß er 'aus Deutschen Landen' kommt, ist wohl auch bemerkenswert: die Firma E-T-A (Altdorf, bei Nürnberg, hiermit engagiert in neuester Technologie), dazu das Fraunhofer Institut für Integrierte Schaltungen (Erlangen), ein Forth Profi (Hamburg), der schon vor Jahren mit den allerersten ersten Forth-Chips befasst war, und Siemens (München) als Silizium-Veredler ergeben einen interessanten Cocktail.

Entwurfsziele

- Kurze Instruktionszyklen
- Deterministisches Zeitverhalten
- Effiziente Interruptverarbeitung / effizienter Taskwechsel
- Für Echtzeitanwendungen optimierter Instruktionssatz

Stichworte:

Forth-Prozessor
Interrupt-Latenz
Task-Wechsel
RISC
Echtzeit

- Hardwareressourcen für die digitale Bild-/Signalverarbeitung

Der FRP 1600 wurde in einer 1 µm-CMOS Technologie gefertigt und arbeitet mit einer Taktfrequenz bis 15 MHz. Durch die spezielle Architektur entspricht dies einer Prozessorleistung von 13 MIPS. Da der Prozessor weder einen internen Programm- oder Daten-Cache, noch eine BefehlsPipeline besitzt, wird eine zeitvariante Befehlsausführung vermieden. Daher verhält sich der FRP 1600 in allen Situationen vollkommen deterministisch. Dies ist eines der wichtigsten Merkmale in Echtzeitsystemen, da somit auch das zeitliche Verhalten des Prozessors bereits während der Programmierung eindeutig vorhergesagt werden kann. Die hohe Verarbeitungsleistung, die sehr schnelle Reaktionen auf externe Ereignisse, der extrem schnelle Task-Wechsel und das deterministische Verhalten machen den FRP 1600 zu dem prädestinierten Mikroprozessor für zeitkritische Echtzeitanwendungen und Aufgaben der Automatisierungstechnik. Damit ist der FRP 1600 in diesem Aufgabenbereich deutlich leistungsfähiger als andere Prozessoren.

Der FRP 1600 im Vergleich

Im Vergleich zu anderen Prozessoren, die einer ähnlichen Entwurfsphilosophie folgen (NC 4000, RTX 2000), ist der FRP 1600 besser für zeitkritische Echtzeit- und Interruptverarbeitung

optimiert und besitzt eine Busstruktur, die zum Industriestandard der 68k-Familie kompatibel ist.

Anwendungsfelder

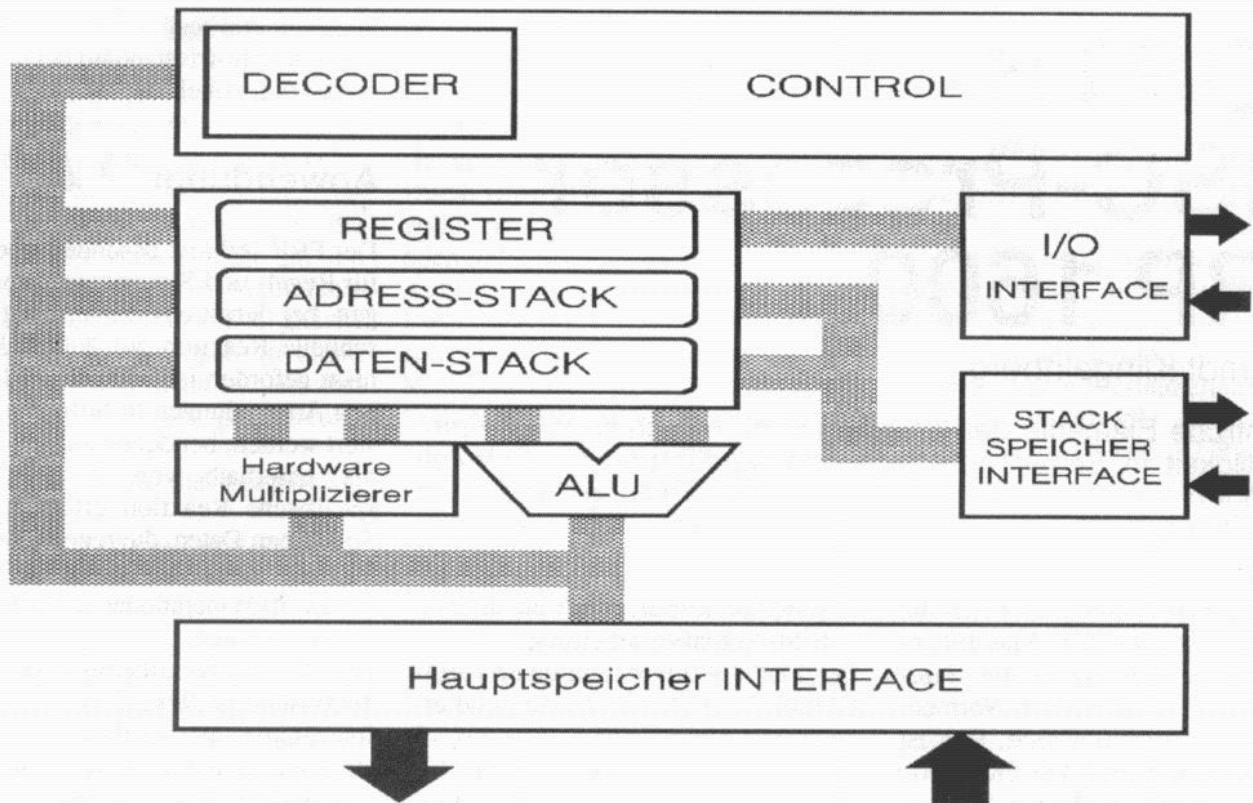
Der FRP 1600 ist besonders geeignet für Regel- und Steuerungsanwendungen, bei denen eine zuverlässige und schnelle Reaktion auf äußere Ereignisse gefordert ist. Mit dem FRP können Anwendungen in Software realisiert werden, bei denen auf ein Ereignis innerhalb von 2 µsec eine spezifische Reaktion erfolgen muß. So können Daten, die regelmäßig alle 50 µsec "eintreffen", vom FRP 1600 mit ca. 700 Operationen je Datum verarbeitet werden.

Die Interruptverarbeitung des FRP 1600 ist besonders effizient: In vielen Interruptroutinen muß nur das FLAG-Register "gerettet" werden, die meisten anderen Daten liegen ja auf den zwei Stacks. In diesem Fällen bietet der FRP 1600 eine Interrupt Latenz nur 2 µsec, realisierbar in Software ohne spezielle zusätzliche Hardware. Ein Taskwechsel kann auf dem FRP 1600 in weniger als 5 µsec realisiert werden, da jede Task im externen Stackspeicher einen eigenen Bereich belegt und für jede Task nur "ihr" Speicherbereich aktiviert zu werden braucht (einfaches Umschalten der Stack-Adressbereiche). Werden spezielle Features des FRP 1600 in den Tasks unterschiedlich genutzt, so sind dann natürlich auch die Index- und andere Register zu retten.

Forth- Stack- RISC- CISC

Die günstigen Ergebnisse von FORTH bei der Realisierung von Echtzeitsystemen führten zu der Entwicklungsvorgabe, den Instruktionssatz des FRP 1600 so zu gestalten, daß FORTH im wesentlichen der Assembler des FRP 1600 ist. Zwangsläufig muß damit der FRP 1600 einer reale Implementation des virtuellen FORTH-Prozessors nahe kommen.

Das hat zu einem Prozessor geführt, der die schnelle Ausführungsgeschwindigkeit eines RISC-Prozessors mit dem semantischen Gehalt von



CISC-Prozessor-Instruktionen verbindet.

Die StackArchitektur ist eine noch weitere Vereinfachung der RISC-Architektur: In den Instruktionen der RISC-Prozessoren gibt es ebenso wie bei den CISC-Prozessoren eine Vielzahl von möglichen Operandenadressierungsarten, die Adressierungsmodi. Diese entfallen bei der Stackarchitektur: Die Daten, die verarbeitet werden, liegen immer auf dem Datenstack. Das vereinfacht die Dekodierungslogik des Prozessors, beschleunigt die Verarbeitung der Daten und macht deshalb ein Pipelining von Instruktionen überflüssig. Der Instruktionscode wird kompakter, so daß mehr interne Hardwareressourcen parallel programmiert werden können. Der Daten- und Returnstack des FRP 1600 ist physikalisch als autonomer Speicher parallel zum Hauptspeicher angelegt (gleichzeitige Zugriffe auf die Stacks und übliches Memory). Deshalb kostet ein Unterprogrammaufruf auf dem FRP 1600 nur einen Zyklus: Das Holen der ersten Instruktion des Unterprogramms aus dem Hauptspeicher findet zeitgleich mit dem Ablegen der Rücksprungadresse auf dem Returnstack statt. Der Rück-

sprung aus dem Unterprogramm findet parallel zur letzten Datenoperation im Unterprogramm statt und wird über das "Return-Bit" im Instruktionscode gesteuert.

Ein Unterprogrammaufruf kostet auf dem FRP 1600 nur EINEN ProzessorZyklus!

Das prädestiniert den FRP 1600 für modulares strukturiertes Programmieren und das wiederum ergibt:

Effiziente Programmerstellung und -wartung.

Hauptkostenvorteil beim Einsatz des FRP 1600 ist seine Programmierbarkeit: Der Assembler ist FORTH, eine höhere Programmiersprache. Und jedes eingebettete System kann als Betriebssystem über ein eigenes Forth-System verfügen, so daß zur Wartung das System auch im laufenden Betrieb (über eine serielle Schnittstelle) umprogrammiert/ent-bug-gt werden kann.

Weitere Details

- Speicher:
 - Hauptspeicher: Adressraum 16 bit CELL-adressiert,

20 bit breite Adressen (16 * 128 KByte)

1 M-'CELLS' (2 MByte)
 Page-weise mit oder ohne Wait-state

- ReturnStack und Paramterstack: Adressraum (beide zusammen) 9 bit also jeweils 256 CELLS (16 bit)
- 20 ns Zugriffszeit für 15 MHz Takt
- Port: 16b parallel
- Indexregister: 3* 4+16 bit breit
- Hardwaremultiplizierer 16b * 16b = 32b (1 Zyklus= 67ns)
- Division 32b / 16b in 1.5 µs (signed oder unsigned)

Quellen: Zusammengestellt aus:
 [1] Presseveröffentlichungen von Norbert Schumann und Gerhard Erbacher, Fraunhofer Institut für Integrierte Schaltungen, Erlangen.
 [2] nach Aufzeichnungen des Autors bei Forth Kongress Rostock 92mar28 Vortrag von Klaus Schleisiek-Kern zu FRP1600.
 [3] Datenblatt von E-T-A, Altdorf (b. Nürnberg)
 [4] Prospekt zur Echtzeit'92 "Tutorial/Seminare" Dr. Konrad W. Koller, Siemens AG



Forth und der Rest der Welt

Kolumne von Andreas Goppold

Wie es sich so ergibt, erreichte mich der Brief von Rolf Kretzschmar, in dem er mir vorschlug, doch wieder mal Beiträge für die Vierte Dimension zu schreiben, just an dem Tage, nachdem ich von der Echtzeit- Messe in Sindelfingen zurückkam, wo ich mein Leibniz-System ausgestellt hatte. Es sind jetzt acht Jahre, seit ich meine ersten Artikel in der Vierten Dimension

Wie stellt man seine speziellen Methoden einer Welt dar, die gewohnt ist, nach ganz anderen Kriterien zu arbeiten und zu urteilen?

veröffentlicht hatte, und seit ich in einem dieser Artikel auch das Projekt Leibniz vorstellte. Acht Jahre sind ein ganzer Lebensabschnitt für einen Menschen, und in der Computerindustrie ein ganz neues Zeitalter. Es haben in dieser Zeit gewaltige Umschichtungen stattgefunden. Spricht man etwa von den 64 K Adressraum eines Microcomputers, mit denen wir damals arbeiten mußten, denkt man an Geschichten aus der Steinzeit. Die Situation der Softwarewelt hat sich radikal gewandelt, Leibniz ist inzwischen in der Version 18.3., ist etwas völlig anderes, als z.B. in jenen alten Artikel aus dem Jahre 1984 beschrieben wurde, es sogar völlig anders, als noch vor einem Jahr. Nur Forth scheint sich recht wenig gewandelt zu haben. Noch immer 1024-Byte Blocks, noch immer DUP ROT SWAP DROP, und noch immer der Griff zum Reset-Knopf (was ist die typische Handbewegung des Forth-Programmierers?). Ich habe mit den Abläufen in der

Forth-Szene seit einigen Jahren nicht mehr so viel zu tun, seit ich ich eingesehen habe, daß die Geschicke von Forth von viel zu vielen widerstreitenden Positionen beeinflusst werden, die in keiner Weise mit der meinigen zu vereinbaren sind, und ich mir schon selber das schaffen muß, was ich haben möchte, und Forth gibt ja dafür auch alle Möglichkeiten. Ich habe mir mit Leibniz ein System geschaffen, das diese Ansichten und Einsichten beinhaltet.

Da ich aber auf der Basis von Forth aufbaue, und ich nach wie vor die in Forth enthaltenen Prinzipien für ungeheuer mächtig und wertvoll halte, denke ich, daß ich dem Geist dieser Zeitschrift immer noch verwandt bin, und ich denke, daß ich auch einige wertvolle Anregungen und Stoff zum Nachdenken bieten kann. Ich tummele mich ja seit einigen Jahren in der akademischen Welt, setze mich täglich mit objektorientierter Programmierung auseinander, habe meine liebe Not mit diesem furchtbaren Software-System namens UNIX, von dem alle Welt glaubt, daß es das Betriebssystem der

Zukunft ist (ich glaubs ja auch, aber nur mangels besserer Konkurrenz. Alle anderen Betriebssysteme, die ich kenne, sind noch viel schlimmer als UNIX). Ich habe also einen intensiven Kontakt "mit dem Rest der Welt" während ich immer noch weiter meine Hauptarbeit mit einem Software-System mache, dessen Abstammung von Forth ich zwar nicht an die große Glocke hänge, um mir nicht unnötig die Chancen zu verderben, aber unter Eingeweihten kann man ja schon mal ein oder zwei Worte diesbezüglich fallen lassen.

Da die Redaktion der *Vierten Dimension* gerade gewechselt hat, stehen alle Weichen auf einen neuen Start, neue Perspektiven, vielleicht das Zurückkommen aus toten Ecken und hohlen Gassen, das Aufpolieren von blinden Flecken und eingetrampelte Pfade Verlassen. Ich weiß noch nicht so recht, worüber ich schreiben werde, und würde mich über Anregungen und Fragen aus der Leserschaft sehr freuen. Ich werde auf keinen Fall über die neunundvierzigste Version berichten, wie man lokale Variablen implementieren kann, noch werde ich mich darum kümmern, wie man durch Ausnutzen einer bisher unbekanntem Flag im 808x Assembler bei einer DO...LOOP noch 3 Microsekunden pro Schleifendurchlauf sparen kann, oder wie man durch geschicktes Platzieren von "--" noch mehr Code in einem 1024-Byte Block unterbringen kann.

Meine Themen sind eher: Was sind die besonderen Denk- und Arbeitsweisen, die jemand, der mit einer erweiterbaren, interaktiven Programmiersprache gearbeitet hat, beherrscht, die andere, die in konventionelleren Systemen arbeiten,

Den Lesern der Vierten Dimension, die von Anfang an dabei waren, ist der Name Andreas Goppold (Dipl.-Inform., Master of Science EDV-Beratung / Software-Design, Unterherrnhäuser Str. 13, D-8196 Eurasburg, Tel (+49) +8179 1479) sicher ein Begriff. Er gehörte zu den Gründungsmitgliedern der Forth-Gesellschaft und veröffentlichte in den ersten Ausgaben der Zeitung einige außergewöhnliche Beiträge zu Forth, Programmiersprachen im allgemeinen, und philosophischen Themen aus dem Bereich Informatik. In den letzten Jahren hat er sich ein wenig aus der Forth-Szene zurückgezogen. Er arbeitete in dieser Zeit an Industrieprojekten und entwickelte sein eigenes universales Software-System Leibniz, das auf den Prinzipien von Forth aufbaut. Er ist jetzt neben seiner freiberuflichen Arbeit an der Universität der Bundeswehr in München tätig, wo er das "Forschungsprojekt Leibniz" leitet. Jetzt meldet er sich zurück, mit einigen neuen Ergebnissen und einer frischen Ansicht zu immer aktuellen Themen.

nicht so gut können? Wo sind die Märkte für interaktive, erweiterbare

Es ist ganz gut, sich ein wenig an der Realität zu orientieren, und Phantasien zu vergessen, die nicht zu realisieren sind.

prototyp-orientierte Systeme, und wie plaziert man sich in solchen Märkten? Wie stellt man seine speziellen Methoden einer Welt dar, die gewohnt ist, nach ganz anderen Kriterien zu arbeiten und zu urteilen? Kurzum, wie positioniert man sich als jemand, der ein bißchen anders ist als die anderen, mit dieser Andersheit, so daß die Anderen merken, daß es nicht immer das Beste ist, "Gleicher zu sein als die Gleichen", sondern daß es ganz allgemein dringend nötig ist, auch ein paar Exemplare von der "ausgefallenen Gattung" bei der Hand zu haben.

Ich möchte mich eigentlich auch nicht so sehr auf "Forth" als Kontrapunkt zu dem erwähnten "Rest der Welt" fixieren, schon allein deswegen, weil es bei 40 Forth-Programmierern 40 total verschiedene und inkompatible Meinungen darüber gibt, was Forth eigentlich ist. Auf solche Diskussionen will ich mich überhaupt nicht erst einlassen. Wenn man mich läßt, werde ich hier einen Sprachgebrauch entwickeln, wie man das mit einer Terminologie, die auch im informatischen Feld mehr den Standards entspricht, besser anfassen kann. Forth ist ein recht ausgefallener Spezialfall einer sehr breiten Klasse von logischen Maschinen, den Tokenlisten-Prozessoren, und es gibt im Compilerbau Zwischencode-Systeme, die Forth sehr ähnlich wären, wenn sie nur jemals ein Benutzer zu Gesicht bekäme. P-Code und die Run-Time Codes von Smalltalk-Systemen sind so aufgebaut. Wie man es in den Wald ruft, so tönt es heraus, und wenn man sagt, man arbeitet mit einem Tokenlisten-Prozessor, dann glaubt der Gesprächspartner wenigstens nicht gleich, jemand vor sich zu haben, der einem nicht nur das Seelenheil als

Programmierer, sondern auch sonst die Lösung aller Probleme, der bekannten, wie der noch nicht geahnten präsentieren will (Forth ist die Antwort, was war die Frage?)

Es ist ganz gut, sich ein wenig an der Realität zu orientieren, und Phantasien zu vergessen, die nicht zu realisieren sind. Der Mainstream der Software-Technologie geht in anderen Bahnen als die von Forth.

In unserem Institut arbeiten wir viel mit dem Begriff "Synergie", und wir haben natürlich unsere Spezial-Bedeutung von diesem Begriff, die auch noch etwas anders ist, als wie es die Firma meint, die da sagt: "Synergy at Work". Synergie bedeutet für uns (unter anderem), daß es

gerade die unoffensichtlichen Mitglieder einer Gruppe sein können, die wesentlich zu dem Erfolg der Gruppe beitragen. So z.B. die Frau, die allen Kaffee kocht, oder der Kollege, der immer herumgeht, an keinem Projekt wirklich zu arbeiten scheint, und immer mit allen Leuten quatscht. Die Welt steht noch vor der revolutionären Erkenntnis, daß ein gewisses Quantum an Vielfalt, ein kleines Quäntchen Chaos, für das ja Forth und die Forth-Programmierer so berühmt sind, mit dem richtigen Management Wunder wirken können. Man muß nur wissen, Wo, und Wieviel,

Der Mainstream der Software-Technologie geht in anderen Bahnen als die von Forth.

und Wann man aufhören muß. (Die drei Großen W's des Software-Managements)

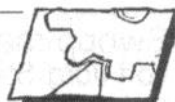
Fragen beantworten...

Forther sind neugierige Menschen. Wir, die Direktoren der FG und die Redakteure der VD, sind Forther. Um den Wünschen der FG-Mitglieder gerecht werden zu können, müssen wir die Wünsche erst einmal genauer kennen! Darüber hinaus möchten wir wissen, mit welcher Hardware sich die FG-Mitglieder so herumschlagen (wollen/müssen). Wenn man so neugierig ist, liegt es nahe, einen entsprechenden Fragebogen an alle Mitglieder zu verschicken. Darum liegt diesem Heft ein Fragebogen bei.

Themenwechsel! Es gibt so'ne und so'ne Mitglieder. Die einen sind die aktiven, die dafür sorgen, daß die FG wächst und gedeiht und daß unsere gelieb-

te VD dick und rund wird. Denen würden wir gerne ein kleines materielles Dankeschön zukommen lassen (vonwegen Motivation und so...). Somit haben wir beschlossen, unter den Autoren der letzten beiden VD-Jahrgänge und allen Einsendern des Fragebogens kleine Geschenke zu verlosen. Diese Geschenke sind z.B. Forth-Bücher in deutscher und englischer Sprache, Leerdisketten, Diskettenboxen oder Fraktal-Programme für den PC. (Weitere Spenden für den Geschenketopf werden dankbar angenommen!) Viel Spaß mit Forth und viel Glück bei der Verlosung wünschen die Direktoren der FG und die Redakteure der VD, die leider von der Verlosung ausgeschlossen sind.

... Preise gewinnen



Swopp mal wieder

von Jörg Staben

Sie wollten sicher immer schon mal die beiden parallelen Schnittstellen Ihres PC per Software vertauschen, oder? Hier ist die Lösung in Forth. Zwar ist die vorgestellte Lösung systemspezifisch (F-PC/TCOM), doch ist sie leicht übertragbar, wenn das jeweilige Forth über Operatoren für den Zugriff auf segmentierte Adressen besitzt.

Ich mag dieses Programm, weil das Vertauschen der parallelen Schnittstellen LPT1 und LPT2 eine richtige DOS-Anwendung außerhalb unseres kleinen Forth-Mikrokosmos darstellt. Das Problemchen, die beiden parallelen Schnittstellen vertauschen zu müssen, ohne die Anschlußkabel aus dem Rechner zu ziehen, ist zwar nicht brandaktuell, taucht aber doch immer wieder mal auf.

Grundsätzlich läßt sich diese Aufgabe lösen, weil die Portadressen der ersten parallelen Schnittstelle LPT1 (378hex) und der zweiten Schnittstelle LPT2 (278h) in BIOS-Variablen eingetragen sind. Diese BIOS-Variablen kann man auslesen, vertauschen und zurückspeichern.

F-PC

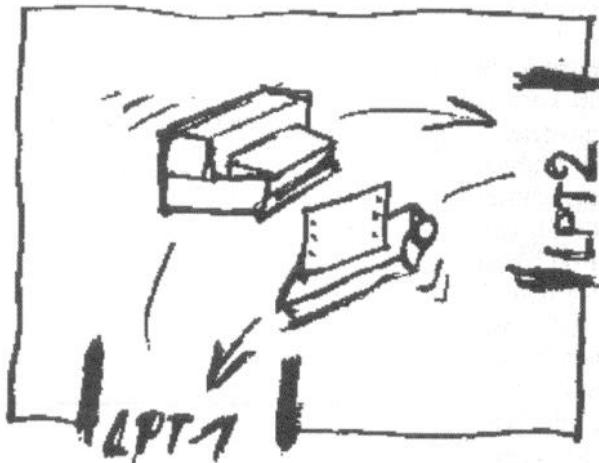
Im F-PC nutzt man natürlich den Dialog mit dem Interpreter, die Stärke von Forth. Prüfen kann man den BIOS-Variablen-Bereich so:

```
hex (dann sehen d. Adressen wie im Buch aus)
lpt1 10 ldump
```

Bei einem Rechner mit nur einem Gerät (meist ein Drucker) an einer parallelen Schnittstelle zeigt der Speicherauszug (dump) an, daß für LPT1 der Wert 378h eingetragen wurde, dagegen die Variablen für LPT2..LPT4 den Wert 0 enthalten. Die Belegung der einzelnen Schnittstellen läßt man

sich sinnvollerweise noch zusätzlich einem System-Analyseprogramm wie Norton's SysInfo, PC-Tools SI oder eben hier SNOOP (shareware) anzeigen, um zu hören, was andere Leute dazu sagen.

Trägt man mit
hex 278 lpt2 !!
von Hand den Wert 278h für die zweite parallele Schnittstelle ein, so meldet Norton's SI beide LPT's mit den ein-



getragenen Adressen, SNOOP ist dagegen noch etwas pfiffiger und sagt, es sei eine parallele Schnittstelle physikalisch installiert, aber zwei LPT's im BIOS eingetragen. Die beiden eingetragenen parallelen Schnittstellen sind mit LDUMP sehr schön zu sehen.

Nach dem Compilieren im F-PC wird erstmal - vor allem anderen - das Wort aus dem Debugger heraus aufgerufen: dbg lptswap

Man sieht, wie die segmentierte Adresse 40:08h richtig als 40 und 8 auf den Stack gelegt wird und wie der richtige Wert 378 ausgelesen wird. Das gleiche geschieht mit 40:0A und

278 - ganz wunderbar!

Nun bestätigt der Check mit SNOOP nach einem LPTSWAP, daß diese Anwendung funktioniert. Ein solches Dienstprogramm läßt sich von der F-PC-Oberfläche heraus bequem mit
sys snoop e
aufrufen.

TCOM

Um eine - von der Größe her tragbare - DOS-Applikation zu bekommen, soll das Programm vom Targetcompiler TCOM übersetzt werden; die TCOM-Programme lassen sich bekanntermaßen bis auf ca. 40 BYTE verkleinern, aber auch ohne Handoptimierung werden generell sehr kleine Programme generiert.

Auf den ersten Blick meint man, ein solch' simples Programm wie LPTSWAP erfordere weder eine An-

passung an TCOM noch weiteres Nachdenken, so daß sofort kompiliert werden kann. Daß die Funktion des Programmes wieder getestet wird, ist selbstverständlich - SNOOP muß also wieder ran.

Arbeitet man mit ZZ, der TCOM-Umgebung, so sollte man sich SNOOP oder sein persönliches, entsprechendes Dienstprogramm in die Datei ZZ.CFG eintragen; damit ist die Kontrolle der parallelen Schnittstellen aus der

TCOM-Umgebung heraus bequem möglich.

Aber zurück zu LPTSWAP:

SNOOP zeigt zum allgemeinen Einsetzen an, daß die LPT's wider Erwarten nicht vertauscht werden!! In einer solchen Situation ist es beruhigend, das F-PC aufrufen zu können.

Je nach Version des F-PC werden Sie sich die integrierte Umgebung aus F-PC-Interpreter und TCOM-Compiler mit einem der externen Editoren SZ oder NEWZ aufgebaut haben, so daß Sie nun jederzeit aus dem Editor heraus mit F5/Ctrl-F5 den TCOM oder mit F9/Ctrl-F9 das F-PC starten kön-

Stichworte

F-PC	3.56
TCOM	2.24
Fehlerkorrektur	
PC	
Anwendungsprogrammierung	

nen.

Der Schlüssel zu solch einer multifunktionalen Umgebung sind die Dateien ZZ.CFG, in der die von ZZ auszuführenden DOS-Kommandos stehen und F-PC.CFG, der dem F-PC verrät, wo es seine Dateien findet.

Hat man das F-PC als Interpreter in die TCOM-Compiler-Umgebung eingebunden, muß FPC.CFG zur Verfügung stehen und FPATH für das F-PC voll qualifiziert sein, d.h. immer mit ...fpc\src\..., damit der F-PC-Komfort auch unter ZZ noch funktioniert (BROWSE und HELP).

Nun wird der F-PC-Ratschlag von oben auch für TCOM beherzigt: Bevor das Programm zum ersten Mal ausgeführt wird, wird der symbolische Debugger (meist über F7 / Ctrl-F7) aufgerufen. In gleicher Weise, wie im F-PC die Werte auf den Stack gelegt und wieder heruntergenommen wurden, müssen sich die Werte jetzt auch im Stackfenster wiederfinden. Das Stackfenster läßt sich mit ALT-S zu- oder wegschalten.

Beobachtet man die Werte, so sieht man, daß die 2VARIABLEN korrekt ihre Werte ablegen, nur die ausgelesenen Adressen der BIOS-Variablen stimmen nicht. Die Werte dieser Adressen sind ja bekannt (378h und 278h) und können im F-PC mit LDUMP jederzeit überprüft werden.

Kommt LPTSWAP zu der Stelle, wo die Werte zurückgeschrieben werden, zeigt sich Forth von seiner besten Seite: CTRL-ALT-DEL oder die RESET-Taste ist angesagt. Nach vielen Testläufen, einem Blick in die TCOM-Library, wo @L definiert ist und dem Vergleich mit dem @L des F-PC fiel nur auf, daß beide Wörter unterschiedlich codiert sind. Aber wer zweifelt schon an Tom Zimmer?

Die Klemme

Dann endlich, eine gute Dreiviertelstunde später, die Idee: Tom wird doch wohl nicht aus Versehen...?? Doch! @L im TCOM erwartet seine Parameter in der Form ADDR:SEG, also verkehrt herum gegenüber dem F-PC und entgegen der Dokumentation der TCOM-Library.

Listing zu: Swopp mal wieder (J. Staben)

```
\\ Vertauschen = swapping der Druckerports LPT1 und LPT2

Compileroptionen nach Standardvorgabe der Umgebungsvariablen TCOM

      SET TCOM= ...

Version:
      1.5   jrg  13.07.92 04:37
      1.1   jrg  12.07.92 12:38
      1.0   jrg  11.07.92 09:55

Compiler:      F-PC 3.56
               TCOM 2.12/8086 2.24

Literatur:     Tischer, PC Intern 3.0; S.570 ff.

Algorithmus:   Die Portadressen der parallelen Schnittstellen
               sind in BIOS-Variablen eingetragen.
               Diese lassen sich auslesen, vertauschen und zurückschreiben.

PseudoPASCAL: DummyWord = MEM[ 0040h:0008h ]
               MEM[ 0040h:0008h ] = MEM[ 0040h:000Ah ]
               MEM[ 0040h:000Ah ] = DummyWord

Forth:
      : lptswap      ( -- )
        lpt1 fetch      \ lpt1
        lpt2 fetch      \ lpt1 lpt2
        both port_addresses are on stack
        lpt1 store
        lpt2 store
      ;

Test:         Das ABUNDANCE-Hilfsprogramm SNOOP zeigt mit

               SNOOP E ( = equipment)

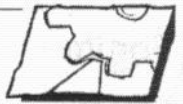
               an, wie die Schnittstellen belegt sind.
               LPTSWAP läßt SNOOP statt LPT1 die LPT2 anzeigen.

               Aufruf aus der F-PC-Oberfläche heraus:
               SYS SNOOP E
```

```
false Value buggy \ Test auf Parameterfehler TCOM v2.24
buggy #IF
)
!!!!!!!!!!!!!! TCOM - Fehlerkorrektur !!!!!!!!!!!!!!!
      @L in der TCOM-Library erwartet nicht SEG:ADDR
      wie angegeben, sondern ADDR:SEG !
!!!!!!!!!!!!!!
{
\tcom : fetch      ( seg addr -- n )      swap @1 ;
\fpc  : fetch      ( seg addr -- n )      @1 ;
\tcom : store      ( n seg addr -- )      swap !1 ;
\fpc  : store      ( n seg addr -- )      !1 ;

#ELSE
      : fetch      ( seg addr -- n )      @1 ;
      : store      ( n seg addr -- )      !1 ;

#THEN
```



Das war's dann schon; ich habe nach einem Weg gesucht, den Quelltext in akzeptabler Form von beiden Umgebungen aus korrekt kompilieren zu können und scheiterte nochmal kurz (wieder CTRL-ALT-DEL) daran:

```
\tcom : @1 ( sg,adr - n) swap @1 ;
```

Das mag TCOM ohne die entsprechende Compiler-Option /RedefOk gar nicht!!

Die Lösung

Weil auch !L davon betroffen ist, war mein erster Ausweg, neue Segment-Operatoren mit einem X davor einzuführen und das war's dann.

Programm fertig, Problem gelöst, eine Forth-Applikation von 2400 BYTE entwickelt; morgen muß ich das noch Tom Zimmer schreiben, denn F-PC und TCOM sollen ja noch schöner werden.

Ach so, in einer Abschluß-Revision habe ich mir den PseudoCode angesehen und mich gefragt, warum sieht dein Forth-Programm nicht so aus? Nun - jetzt sieht es so aus!

Fortsetzung des Listings zu: Swopp mal... (J. Staben)

```
***** jrg 12.07.92 11:54 *****
LPTSWAP
Vertauschen der parallelen Schnittstellen
*****
(
\ipc anew lptswapper

: .title      ( -- )
              10 10 49 17
              box&fill
              bcr ." LPTSWAP vertauscht LPT1 und LPT2"
              bcr ."      LPTSWAP ist FREeware
              bcr ." & eine F-PC/TCOM Anwendung"
              bcr ."      [c] J.Staben 1992"
              bcr bcr
              ;

$0040 $0008 2Constant LPT1
$0040 $000A 2Constant LPT2

: lptswap    ( -- )
            lpt1 fetch      \ lpt1
            lpt2 fetch      \ lpt1 lpt2
            lpt1 store      \ lpt1
            lpt2 store
            .title
            ;
```

Übrigens...

Zum Schluß noch etwas Seltsames aus der Computerwelt:

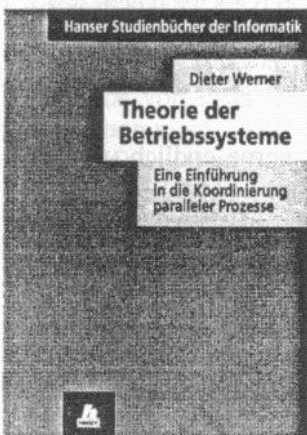
Wie aus gewöhnlich gut unterrichteten Kreisen verlaute, kommt das

PC-Abenteuerspiel Kings Quest V nicht mehr mit einer 20MB Festplatte aus, sondern umfaßt in der CD-ROM-Version fette 180MB - endlich mal ein vernünftiger Kopierschutz.

Buchbesprechung

Theorie der Betriebssysteme Eine Einführung in die Koordinierung paralleler Prozesse

Prof. Dr. mult. Dieter Werner
Carl Hanser Verlag, München, 1992.
ISBN 3-446-16547-9
262 Seiten, 48,00 DM



Das Buch führt in verschiedene Bereiche des Fachgebietes Betriebssysteme ein. Von besonderer Bedeutung sind dabei die theoretischen Grundlagen. Auf gut 250 Seiten und 5 Kapiteln wird der Stoff methodenorientiert dargestellt. Der Leser wird dazu angeregt auch bei Zusammenhängen, die intuitiv als richtig akzeptiert werden, auf Beweisführungen Wert zu legen und zwischen den Aussagen UNWAHRSCHEINLICH und UNMÖGLICH streng zu unterscheiden.

Im einzelnen behandeln die Kapitel folgende Themen:

1. Prozeßsysteme
2. Koordinierung paralleler Prozesse
3. Synchronisation
4. Betriebsmittelverwaltung
5. Verklemmungen

Das Buch ist für das Selbststudium konzipiert und verbindet die formale

Behandlung von Problemen mit Algorithmen zu deren Lösung. Für die Darstellung von Algorithmen wird dabei ein Ada-ähnlicher Pseudocode verwendet.

Obwohl sich das Buch vor allem an Studenten der Informatik im Hauptstudium wendet, ist es sicherlich auch für FORTH-Programmierer interessant, da diese sich oftmals durch Erweiterung eines FORTH-Systems Werkzeuge schaffen, z.B. Semaphore, die die Koordinierung paralleler Prozesse erlauben.

Wer es akzeptiert, sich mit der Programmierung auch theoretisch auseinanderzusetzen, für den ist das Buch wirklich eine Bereicherung.

Heinz Schnitter (hfs)



in Forth !

Angeschlagen von Jörg Staben

Hagelkreuz- 23, W-4010 Hilden, Tel.: 02103 - 240609

DAS! wollen wir in Forth sehen!

Klarstellung: Die neue Rubrik WANTED hat einen nicht ganz eindeutigen Namen: Hier treffen sich nicht nur Hilferufe und Antworten, sondern auch Themenvorschläge, wie SYSINFO einer war.

Jetzt höre ich sie schon wieder schreien, die Negaholics, die ewigen Miesmacher: "Wie trivial, zu einfach, zu simpel, sowas Billiges!" Dennoch werden hier auch Themen präsentiert, zu denen bereits Literatur und Lösungen existieren; damit fällt der Aufwand für die - zum Programmieren immer notwendige - Recherche weg und das Ergebnis kann mit einem fertigen Produkt verglichen werden. Jetzt ganz klar gesagt: Jemand, der sich auf ein WANTED In Forth hin meldet, bekommt die passenden Unterlagen (Zeitschrift/Disk etc.) und ein Forth zugesandt, damit es nicht an den Werkzeugen scheitert.

Geräte-Treiber

Das Thema ist z.Zt. Dauerbrenner in allen DOS-Magazinen, die Arne Schäpers (DOS.5-Spezialist) bezahlen können. EIN Forth kann's schon: Tom Almy's 4THCMP - ein reines Compiler-Forth für kleinste Anwendungen. Aber - wer bringt's dem TCOM bei?

dBASE-Dateizugriff

Daß keiner mehr seine Adreßverwaltung in Forth schreibt, mag stimmen oder nicht. Was aber m.W. jedem Forth fehlt, ist ein dBASE-Dateizugriff.

Wer lehrt das F-PC, mit dBASE III+ Dateien umzugehen?

SYSINFO

Unser eigenes SysInfo haben wir noch nicht; bitte melden!

Phonetische Suche

Ich bekneie schon seit Monaten Tom Zimmer, seine Interpreter-Schleife so zu ändern, daß der innere Interpreter nach FIND und NUMBER im Fehlerfall nicht blöd "?" rumfragt, sondern wild entschlossen eine Vorwärtsreferenz NOTFOUND ausführt.

Sollten meine Überredungsversuche Erfolg haben, wäre mein erstes NOTFOUND eine phonetische Suche, die z.B. nach dem tumben Tippfehler in WODS ganz schnell und pffiffig WORDS zur Ausführung anbietet. Wer sorgt dafür, daß NOTFOUND aus WODS WORDS macht?

Nutzung von Bibliotheken in Forth

Während wir den Nachweis, daß Forth so'ne tolle Programmiersprache ist, immer noch schuldig bleiben, sit-

Gesucht wird . . .

von Rolf Kretzschmar

ZX81-Forth

Für meine Sammlung an Rechnern, die mit Forth als Betriebssystem arbeiten, suche ich noch immer die ROM-Version von David Husband, Skyvave Software, Bournemouth, England, 1983. Ein Handbuch ist als Kopie vorhanden. Übrigens konnte dieses Forth bereits Multitasking! (Wenn das Handbuch nicht lügt, ist das kein Witz!)

Forth Funken

Da Forth als Programmierumgebung für Techniker und Bastler bestens geeignet scheint, müßte es doch auch fortheinnde Funkamateure geben. Welcher Funkamateure hat das Steuerprogramm für drehbare Antennen in Forth? Welcher Funkamateure berichtet über seine Erfahrungen mit Forth? Welcher Funkamateure stellt sich als Ansprechpartner zur Bildung einer Fachgruppe zur Verfügung?

Forth Messen

Für den Unterricht in Meß- und Regeltechnik (Technikerschule) suche ich alles, was mir hilft, Forth einzusetzen. Also: Erfahrungen, Ratschläge, Software, Hardware und Informationen (auch kommerzielle!). Zudem steht in der Redaktion schon ein Ordner mit der Aufschrift: Themenheft X/1993: Messen-Steuern-Regeln. Kluge Firmeninhaber reservieren schon jetzt Platz für Anzeigen!

CFORTH - Erfahrung ?

Ich würde gerne endlich -C- lernen. Wer hat CForth-93 (Bradley Forthware) auf ISA/DOS unter MS-C oder Borland (ohne UNIX) zum laufen gebracht?

Arndt Klingelberg
Tel. 0 +2404 - 61648
Strassburger Str. 12
D-W 5110 Alsdorf



zen die Jungs und Mädels in Kalifornien nicht in, sondern an einer excellenten Bibliothek. An der Universität Riverside UCR wird eine Assembler-Library ständig weiterentwickelt und gepflegt(=getestet) und als pd (public domain) zur Verfügung gestellt.

Diese Bibliothek von Assemblerprogrammen enthält alles, was das Herz begehrt, so daß man mit MASM oder TASM, den Assemblern von Microsoft und Borland, ruck-zuck Programme schreiben kann. Vor allem Programme, die laufen und was machen. Auch von Forth aus würde man gerne mal auf eine Routine wie CPUID.ASM zurückgreifen, um sein Programm prozessorspezifisch zu optimieren. Aber leider kann uns offensichtlich keiner zeigen, wie man prinzipiell ein solches fertig konfektioniertes Assemblerunterprogramm von Forth, vom F-PC aus nutzen kann. Denn die meisten von uns sind gerade damit beschäftigt, die lokalen Variablen zu implementieren. Ja, ich weiß: Hat Tom Zimmer auch gemacht ... aber schon vor vier Jahren.

Resonanzen

Trotz der oben genannten Mißverständnisse hat sich einiges getan:

KEYBOARD-Treiber

Die Zeitschrift DOS TOOLBOX hat in ihrer aktuellen Ausgabe auf mein WANTED reagiert und einen 800 Byte großen Tastatur-Treiber in Assembler vorgestellt (der 2K große JJANO.KBD ist in C). Damit dürfte das Thema erledigt sein, es sei denn, jemand zeigt uns, daß man sowas auch in Forth machen kann.

Interrupts

Friederich Prinz, Moers, hat meinen Wunsch nach einer TSR- Uhr mehr als erfüllt. Exemplarisch zeigt und beschreibt er, was im ZF zu tun ist, um ein speicherresidentes Programm - die UHR - zu programmieren. (Siehe dazu: Uhr-Treffer)

Sie sehen, wir sind alle gleichermaßen von Forth fasziniert. Es ist nicht nur die grenzenlose Sprache und das offene Konzept, von dem diese Faszination ausgeht? Es ist auch das Programmieren selbst, das sich Messen

an Problemen, die Lust an der Lösung!

Treffer gelandet

akg. Unter 'Wanted' wurde in der VD 2/8 nach einem LIFO-wordbuffer für F-PC verlangt. *Jörg Staben* (vielen Dank!) machte mich aufmerksam, daß *Tom Zimmer* hier selbst nun abgeholfen hat. Die Änderung ist in 'whats-new' jedoch nicht dokumentiert worden. Wer die WORD BUFFER Funktionen komplett haben möchte (eben auch mit Wort-Kopie in den Buffer und Mehrfach-Kopie aus dem Buffer) erhält das komplett mit dem Zeilenbuffer gegen SAFU-MFD (SAFU =: selbst adressierter Freiums Schlag, MFD =: mit formatierter Diskette) oder über die Forth-Mailbox. Funktionstasten dafür gibt es aber nur im Modus 'F-new' der ak-Version.

Uhr-Treffer

rk. *Friederich Prinz* hat es sich nicht nehmen lassen, das WANDED-in Forth-Problem mit der residenten Uhr in der für ihn typischen Weise zu behandeln: Der Redaktion liegt von ihm ein 22 Seiten umfassender Aufsatz vor, der auch für den nicht versierten Programmierer die Problematik bei der Erstellung von TSR-Programmen verständlich behandelt. Die Assembler-Programmteile sind nicht einfach mit kurzen Kommentaren versehen, sondern ausführlich beschrieben. *Friederich Prinz* liefert mit diesem Aufsatz ein Gerüst für ähnliche Aufgaben. Ach ja, der komplette Quellcode liegt natürlich auch vor und die Uhr läuft! Freuen Sie sich auf die nächsten VD's, denn wir werden den Artikel aller Vorransicht nach in zwei Teilen abdrucken.

Ahnungen gesucht

von Thomas Hotz,

Metzgerwaidring 25, 7760 Radolfzell, Tel.: 07732-58652

Ich benutze ein F83 Forth (F-PC 3.5) und möchte zum Beispiel ein paar Microcontroller im Netzwerk betreiben können mit der Möglichkeit, daß alle Controller an einer Schnittstelle COM1 oder COM2 hängen. Die ganze Geschichte soll über den PC steuerbar sein, wobei die Schnittstelle im Hintergrund arbeiten sollte. Ich habe allerdings noch keine Ahnung, wie man so etwas macht, da ich erst mit Forth angefangen habe. Hat jemand so etwas schon mal gemacht, oder weiß jemand, wo man etwas über Schnittstellen-Programmierung nachlesen kann?

Ferner suche ich nach guten Büchern über Forth (F83 Standard), in denen auch gezeigt wird, wie man Grafik, Pulldown-Menues, Maus, Schnittstellen und allgemeine Sachen programmiert. Es gibt in der Forth Mailbox einen Lehrgang für den Umgang mit F-PC. Das Problem ist nur, ich habe kein Modem. Kann mir vielleicht jemand Kopien davon zukommen lassen ?? Es wäre echt super, wenn ich ein paar Tips bekäme!!!

Restbestand F-PC-EZ

Anzeige

Spezial-F-PC aus Echtzeit'92 Messe-Angebot, sofort lauffähiges, bereits erweitertes F-PC System inkl. ausführlichen HilfeTexten auf HD-Diskette (5" oder 3", wie verfügbar), DM10 Vorkasse und SAFU (Selbst Adressierter FreiUmschlag).

Arndt Klingelberg
Strassburgerstr.12,
D-W 5110 Alsdorf

Verkaufe...

Anzeige

...original LMI PC/FORTH 3.2, Grafik Erweiterung, Floating Point Erweiterung, Nativ Code Compiler, Binary Overlays, Full Source Code, 300 Seiten Doku.. NP 399,- DM, VP 299,-DM.

Konstantin Vassilev,
Kranichstr. 25,
W-8500 Nürnberg 90
Tel: 0911-32 62 817

F-PC-Grafik

Einheitliche
Grafikschnittstelle verfügbar

js. Obwohl das F-PC die unterschiedlichen Grafikkarten unterstützt, ist eine einheitliche Nutzung der Grafik im F-PC nicht möglich. Während die VGA-Grafik im großen Rahmen zur Verfügung steht und EGA / CGA entsprechend ihrer reduzierten Hardware eingesetzt werden, führt die Hercules-Karte im F-PC ein Schattendasein. Zudem verhindern unterschiedliche Befehle die Erzeugung von Programmen, die auf jeder Grafikkarte laufen. Diese wenig förderliche Vielfalt wird durch die einheitliche Grafikschnittstelle von *Ingo Mathyl* für das F-PC wieder auf einen Nenner gebracht. Um eine maximale Performance zu erhalten, sind nahezu alle Teile des Paketes in Assembler codiert. Das grafikkartenunabhängige Softwareinterface soll 149,- DM kosten. Einige Leistungsmerkmale im Überblick:

- SVGA- und 8514 API- Unterstützung
- Automatische Erkennung des Grafiksystems inkl. Subsystem
- Animation über Colour Cycling
- HiColour (32k-Farben)

F-PC-ak 4.0

Neue Version verfügbar

rk. Arndt Klingenberg mag das F-PC von Tom Zimmer, weil es so mächtig ist. Er hat es verändert, weil er mit zahlreichen Details nicht zufrieden ist und weil ihn *R. Kretzschmar* häufiger bat, das eine oder andere etwas schultauglicher zu gestalten. So wurde schließlich aus dem F-PC ein F-PC-ak. Beim Vergleich der beiden Versionen verhält es sich wohl ähnlich, wie mit DR-DOS und MS-DOS: das eine wird wegen der zusätzlichen Funktionen und Hilfen gerne eingesetzt, dafür hat das andere den Vorzug, über alle Kompatibilitätsprobleme erhaben zu sein. DR-DOS ver-

sucht letzteres durch die höhere Versionsnummer auszugleichen. *Arndt Klingenberg* bietet ab Dezember 1992 seine stark bearbeitete Version als F-PC-AK 4.0 an. Diese AK-Version soll nun kompatibler zum aktuellen Original (V.3.5610) als zum F-PC 3.50 sein. Im Bedienteil wurde die Ähnlichkeit zu Borland/WordStar weiter erhöht. Wirklich erfreulich an diesem F-PC-Derivat ist, daß die Einbindung des deutschen Zeichensatzes und aller 8bit-Zeichen so natlos gelungen ist.

Verwirklicht

Die Spielekiste
für MicroController

js. Auf der letzten Mitgliederversammlung in Rostock wurde der Vorschlag gemacht, eine Mikrocontroller-Ecke in der VD einzurichten. In dieser Ecke sollen die ersten Gehversuche mit diesen aufregenden Geräten gemacht werden und die positiven und negativen Erfahrungen damit ausgetauscht werden. Erfreulicherweise haben bereits mehrere Firmen diese Mikrocontroller-Ecke mit Geräten unterstützt; wir möchten uns an dieser Stelle bei *F.Stüss*, *K.Kohl*/*H.Willers* und *R.Deliano* bedanken.

Diese Mikrocontroller sollen innerhalb der Forth Gesellschaft verliehen werden; ein solches Gerät verbleibt eine bestimmte Zeit bei einem Mitglied, von dort wird es dann dem nächsten Interessenten zugesandt. Der aktuelle Aufenthaltsort der Mikrocontroller ist in der Vierten Dimension unter der o.g. Rubrik zu erfahren. Nun sind das aber alles hehre Ziele; es hat sich in der Diskussion herausgestellt, daß man schon einiges an Vorwissen braucht, um so ein Ding zum Quieken zu kriegen und außerdem quieken die auch gar nicht. Wir benötigen dazu DRINGEND Hilfe von Leuten, die bereits Erfahrung mit der Lösung der Standardprobleme (Stromversorgung, Datenkabel, Datentransfer) haben und diese so aufschreiben, daß andere damit zum Ziel kommen!

So sind wir uns einig, daß es eine Vorstufe für den Umgang mit MikroControllern geben soll. Erfreulicherweise wird in der nächsten VD eine solcher idealer Einstieg vorgestellt, mit dem wir uns auf die ersten Schritte hin zu Messen, Steuern, Regeln konzentrieren wollen.

Zuckerbrot und Peitsche

Leserbrief von Jörg Plewe

Großbaumer Str. 27, 4330 Mülheim a.d.R.,
Tel.: 0208-423514

(...) Zuerst mal will ich es nicht versäumen, Euch zu dem letzten Heft zu gratulieren. So was hat uns nun wirklich lange genug gefehlt. Klasse! Viel, viel Licht!

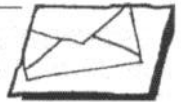
Damit Ihr seht, daß ich das Heft auch gelesen habe, will ich auch noch kurz erzählen, wo ich (noch) Schatten sehe. Direkt aufgefallen sind mir die ak-Kommentare auf den Seiten 12 und 19. Die sind aufdringlich und völlig überflüssig. (...) Ich finde, man sollte einen Redaktionsposten auch nicht ausnutzen, um bei einer solchen Gelegenheit auch noch Werbung für seine eigenen Artikel im Folgeheft, wo dieses Thema dann auch ordentlich bearbeitet wird, zu machen. Der Autor wird immer kleiner und kleiner... Das war also gaaa nix! Genaugenommen war es eine Unverschämtheit.

SORRY!

```
: result-good? ( -- fl )
  source_problems?
  Autor_Urlaub?
  ErstAusgabe_Aachen?
  Zeitdruck?
  and and and 0= ;
```

Habe ich 'P.S.1.' des Autors gar zu wörtlich genommen oder muß jeder Kommentar erst mindestens 3 Monate verzögert werden? O.K., manche ja! Rund um die 'Kompatibilität' würde ich gerne die MeinungsVielfalt unserer Leser kennenlernen.

Arndt Klingenberg (akg)



Brief aus der Provinz

Leserbrief von Friedrich Prinz

Homburger Str. 335, 4130 Moers 1, Tel.: 02841-58398

Die Zeit der großen Sommerferien ist auch in der Moerser Forthgruppe die Zeit des Urlaubes. Alle Aktivitäten sind weitgehend 'eingefroren' (...) und das 'Vereinsleben' findet über Grußkarten aus den Urlaubsorten statt. Wer noch, oder schon wieder, in Moers weilt, denkt an unser Sommerfest zurück, mit dem wir auch in diesem Jahr unseren 'Jahresabschluss' ein wenig gefeiert haben. (...)

Unser neuer 'Kurs' wird ab der zweiten Septemberwoche anlaufen und ist bis ins letzte Detail vorbereitet, so daß ich mich, obwohl auch bereits aus dem Urlaub zurück, noch für ein paar Tage zurück lehnen und die Ruhe in der Stadt genießen kann. Aber ab dem Kursbeginn werden wir uns in der Gruppe intensiv damit auseinandersetzen, "wie FORTH von innen aussieht". Das heißt, daß wir uns ansehen werden, was der Compiler veranstaltet, wenn er Wörter in das Dictionary einbindet, wie die Einträge in das Dictionary aufgebaut sind und wie man Worte intern verfolgt, auch wenn diese wie bei 'unserem' ZF, Code und Wort-Kopf in unterschiedlichen Speichersegmenten ablegen. Das wird ein interessanter Herbst werden. (...)

Die letzte VD hat mich ein wenig geärgert. Da schreibt Jörg Staben in seinem Direktorial davon, daß die immer größer werdenden Applikationen und Entwicklungsumgebungen einen gewissen Unmut hervorrufen. Bravo - laßt uns zum fröhlichen Saurierschlachten blasen! Das ist ganz unsere, Moerser Rede. Um so weniger verstehe ich dann aber, daß ich in der VD eigentlich (fast) nur noch vom F-PC lese. F-PC ist doch auch ein solcher Saurier! Die Moerser Gruppe hat sich das System ein halbes Jahr lang angesehen und dann vehement verworfen - weil es zu groß, zu unhandlich und zu unübersichtlich ist. Wir haben sogar das um teures Geld angeschaffte LMI PCFORTH zur Seite gelegt - aus ähnlichen Gründen - und uns, nach vielen Überlegungen und nach reichlichem Ausprobieren mit unseren ehemaligen

Anfängern, für das ZF entschieden. Dabei werden wir auch bleiben, auch wenn im Augenblick der Anschein dafür spricht, daß das F-PC einen gewissen 'Push' erfahren soll. Die Moerser sind davon überzeugt, daß das F-PC der falsche Weg ist. Deshalb werden wir dieses System nicht tragen und unsere Kurse weiterhin auf dem ZF aufbauen. Das solltet Ihr wissen. (...)

Hier in Moers entstehen laufend neue Sachen, die bei Bedarf von allen Interessenten angefordert werden können. Im Augenblick arbeite ich z.B. an einer Beschreibung des Assemblers in ZF, die sicher "grundsätzlich" auch für andere FORTH-Systeme relevant sein wird. Außer in LMI's Manual habe ich bisher zu keinem FORTH-System eine vernünftige und brauchbare Beschreibung zur Einbindung von CODES gesehen. Dabei läßt sich mit den meisten FORTH-Assemblern

durchaus mehr als nur Primitive wie DUP programmieren.

Diese Arbeit wird allerdings in ihrer 'Reinschrift' frühestens zum Anfang des November fertig sein. Der Redaktion der VD stelle ich auch diesen Text gerne zu Verfügung - wenn daran Interesse besteht. Ansonsten mögen alle Interessenten Ihn zur rechten Zeit bei mir anfordern. (...)

Das war's wieder einmal aus Moers - für heute. Ich wünsche allen Lesern im Namen der Moerser Gruppe einen angenehmen Urlaub verlebt zu haben. Uns allen wünsche ich einen fröhlichen Herbst - und gemütliche Stunden in den Biergärten der Republik.

Moers, den 19.08.'92

P.S. Ein Nachsatz sei mir gestattet: Ein freundlicher Zeitgenosse aus Pforzheim hat mir mit Poststempel vom 6. August ein Couvert mit 5 Stück 3,5" Disketten geschickt. Leider lag diesen Disketten weder ein Brief bei, noch enthielt das Couvert irgendeinen Hinweis auf den Absender. Diesen bitte ich, sich bei mir zu melden und mir mitzuteilen, was er auf diesen Disketten zurück erwartet.



Allerlei von Staben

Leserbrief von Jörg Staben

Hagelkreuz 23, W-4010 Hilden, Tel.: 02103-200609

Liebe Leserinnen und Leser,

als mir in der letzten VD direkt auf Seite 1 mein Konterfei entgegensprang, habe ich gesehen, was Forth aus einem Menschen machen kann. Damit Ihnen nicht Gleiches widerfährt, finden Sie hier einige Anmerkungen zur letzten VD. Der Beitrag LIGHT-SHOW mit seinen beiden Themen "Logikoperatoren" und "Auswerten der DOS- Kommandozeile" hatte einige Ecken, von denen Arndt (Danke!) direkt eine geglättet hat:

□ `clear` hat das Bit nicht gelöscht, sondern umgeschaltet. Bitte ersetzen Sie in `clear` das `xor` durch `not and`.

- Auf S.11 wird `RAND3.SEQ` erwähnt. `RAND3` ist ein leistungsfähiger Zufallszahlengenerator, den Sie im Grafik-Paket von Dr.Smiley finden. Sollten Sie dieses noch nicht ausgepackt haben: `SPLAT`, ein weiterer Geniestreich von Tom Zimmer, enthält ebenfalls einen Random Number Generator und versteckt sich im Archiv `TCOM-SAMP.ZIP`.
- Die Tastaturabfrage auf S.12 muß besser `key upc 'J' #IF ...` heißen.
- In `LOCK_OFF` taucht das Wort `dostib` auf, dem der Ventura Pub die spitze Klammer zu `dos>tib` geklaut hat. Aber mit `words dos` oder `words dos tib` war dieses Mysterium hoffentlich leicht zu lösen.

sen.

Bei dieser Gelegenheit habe ich zu meinem Entsetzen gesehen, daß das Wort dosver im F-PC enthalten ist; ein Wort, das ich gerade programmiert hatte. Tom Zimmer nimmt uns aber auch jede Möglichkeit, am System rumzumachen! Wir werden eines Tages aus lauter Verzweiflung noch Programme damit schreiben...

- Die Datei LOGIC.SEQ sollte in dieser Ausgabe zu finden sein.
- Es gibt im F-PC ein TURNKEY, das ca.60KB in der Anwendungsgröße einspart: ' <name> is boot turnkey <programm.name>'
- Folgt man der Empfehlung von akg und opfert die Compilersteuerungen \fpc und \tcom der Kompatibilität mit veralteten F-PC-Versionen oder dem AKG-Forth, verliert der Quelltext deutlich an Qualität. Denn gerade diese putzigen Wörter zur Compilersteuerung erlauben es, EIN- und DENSEL-BEN Quelltext sowohl vom F-PC als auch von TCOM übersetzen zu lassen! Da wohl kaum jemand über ein vernünftiges Konzept zur Quelltextverwaltung oder Versionskontrolle verfügt, sind diese Compilersteuerungen ein praktikabler Weg, die Arbeit an einem Thema konsistent zu halten.
- Wohl dem, der ein unverändertes, originales F-PC benutzt: Er oder sie kann die Form des Kommentars so nutzen, wie sie der Amerikaner Glen Haydon in seinem Beitrag BRACES.SEQ als neuen Stil der Dokumentation vorgestellt hat. Ich halte es zudem für überflüssig, neben comment: und comment; oder } und { noch weitere Kommentarformen einzuführen.
- Um das Argument für neue Kommentarformen zu entkräften: Steht man vor dem Problem, EIN- und DIESELBE Datei einmal mit einer Demo zum Testen, ein anderes Mal ohne diese Demo im BATch laden zu lassen, dann beenden Sie bitte im Quelltext das Compilieren mit \s, wie Sie es gewohnt sind. Im BATch bricht der Compiler dann an dieser Stelle ab. Um die Tests, die NACH \s folgen, auszuführen, lehnen Sie sich nun entspannt zu-

rück und sagen folgendes:

```
switch noop \s
(\s soll nix machen! )
F-PC sagt: 'Oh, NOOP is' Code!
Nun gut, Sie sagen:
: noop ;
switch noop \s
(jetzt macht \s nix! )
Ich diskutiere nicht mit Computern,
ich programmiere sie!
```

- Ich habe Verständnis dafür, daß manche dem anscheinend unwiderstehliche Drang, ein bestehendes System zu verändern, nachgeben, doch sollte dies bitte im stillen

Kämmerlein geschehen. Zudem bewahrt ein neuer Name die unschuldigen Originale davor, mit einer falschen Kopie verwechselt zu werden.

- Vor allem - lassen Sie sich nicht von der martialischen Checkliste für Autoren abhalten; schicken Sie der VD, was Sie interessant und wichtig finden. Vielleicht wird's im Dialog mit der Redaktion noch runder, vielleicht sagen Sie: "Nee, das ist ok so!"

Boshafte Betrachtung

Leserbrief von Robert Freitag,

Edelweißweg 9, 5000 Köln 71

Sehr geehrte Damen und Herren, es klang schon öfter in der VD an, daß man Interesse habe, den Anwenderkreis der Forth-Nutzer zu erweitern. Zwingend notwendig ist dazu ein Forth System, in meinem Falle F83. Und dann?

In der Dokumentation dazu sind die Befehle View, Show, Fix und See erwähnt. Außerdem wird angedeutet, daß ein Shadowscreenkonzept unterstützt wird, welches durch die Anweisung A erreicht wird. Wie man den Shadowschirm verlassen kann, steht dort leider nicht.)

Das Austesten der o.g. Befehle ergab bei See immerhin, daß man nachvollziehen kann, wie sich ein Befehl zusammensetzt. Das Ende meiner ersten Sitzung erreichte ich während des Probierens des zweiten Befehles: obwohl das gesamte System auf C: liegt, greift VIEW immer auf A: zu.

Es ist mir bisher verborgen geblieben, wie sich unter diesen Umständen der Kreis der Forth-Programmierer vergrößern soll. Programmierer, die die ersten Schritte in dieser Sprache unternehmen, neigen stark zur experimentellen Programmierung.

Bei sehr boshafter Betrachtungsweise könnte man annehmen, daß das gewollt sei. Einige Ursachen liegen auf der Hand: es ist bei meinem System kein Befehl vernünftig dokumentiert und es existiert kein Tutorial.

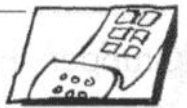
Eine Problemlösung könnte ein Anfängerforum sein. Dieses Forum sollte auf Anfragen aber schnell reagieren. Ein weiterer Ansatz kann eine Art Lexikon sein, in welchem für Anfänger alle Standardwörter erläutert werden und jeder Programmierer seine Worte erklären kann. (Dieser Gedanke ist im Ansatz von J. Staben in der VD 1/92 umrissen worden).

Eine Programmiersprache ist nicht nur gut, wenn sie guten Quellcode erzeugt und einfach zu handhaben ist, sondern sie muß auch im Alleingang erlernbar sein. Leider ist von dieser Eigenschaft (noch?) nichts zu sehen.

Streicheleinheit

"(...) PS: Irgendetwas hat sich geändert an der 'Vierten Dimension'. Ohne genau sagen zu können, was, ist mir aufgefallen, daß ich zum ersten Mal fast das ganze Heft gelesen habe."

(Über dieses post scriptum von Herrn Josef Rieger aus Gaggenau hat sich die Redaktion besonders gefreut. Vielen Dank auch für das zugesandte Foto! rk)



Forth-Vertreib

Es gibt ihn noch:
den Forth-Vertrieb

rk. *Johannes Teich* sitzt immer noch am offiziellen Forth-Gesellschaft-Vertriebs-Kanal! Was es von dort zu Zeit alles abzusaugen gibt, sollte demnächst an dieser Stelle aufgelistet werden. Da das nämlich kaum jemand weiß, hat Johannes Teich in letzter Zeit wenig zu tun auf seinem Kanalposten. Er schaut also in die Röhre. Seine Adresse:

Ettaler-Mandl-Weg 19
W-Murnau
Tel.: 08841-2943



Forth out?

Zur neuen Kolumne auf Seite 25

rk. "Die Konzeption von Forth ist nicht mehr zeitgemäß. Leibniz ist ein Ansatz, die besten Ideen von Forth in das Zeitalter der objektorientierten Programmierung auf multi-Mips und Multi-Megabyte Workstations hinüberzubringen. Ob sich eine solche Herangehensweise bei den Forth-Fundamentalisten durchbringen lassen wird, ist nicht mehr meine Frage." schreibt *Andreas Goppold* der VD-Redaktion. In der gerade fertiggestellten Leibniz-Version 18.0 findet sich kein Hinweis auf Forth. Immerhin konsequent. Wer schon länger die deutsche Forth-Szene beobachtet, wird sich an manchen herausragenden Aufsatz von *Andreas Goppold* erinnern. Daß seine Anmerkungen zum Widerspruch reizen, kann durchaus positiv gesehen werden. Unter dem von ihm selbst gewählten Titel *Forth und der Rest der Welt* hat er sich nun in der VD zurückgemeldet. Wenn er und die Leser es wünschen, könnte daraus eine Kolumne werden. Querdenker sind gefragt! Willkommen an Bord!



Forth in Mailboxen

Auf Anregung des FoeBuD e.V. (Verein zur Förderung des öffentlichen bewegten und unbewegten Datenverkehrs, c/o Art d'Ameublement, Marktstr. 18, W-4800 Bielefeld 1, Tel.: +49 +521 -17 52 54) (MailBox: BIONIC) eine Auflistung von Mailboxen in Deutschland, in denen die Forth-Nachrichten erhältlich sind. Die Zusammenstellung besorgte Johannes Teich (Forth-Vertrieb).



Z-Netz, Brett: /FORTHEV/FORUM

AME	Erlangen	09131- 992998
BIONIC	Bielefeld	0521- 171188
CBRA	Murnau	08841- 5870
INFINET	Muechen	089- 656632
KBBS	Kiel	0431- 336038
RODERICK	Meerbusch	02132- 80884

MausNet, Gruppe FORTH

AC	Aachen	0241- 500482
AC3	Aachen	0241- 514646
A-W	Wien	0043- 222-7984204
BN	Bonn	0228- 213230
COE	Senden/Coesfeld	02536- 8583
DU	Moers/Duisburg	02841- 55275
HB	Bremen	0421- 86675
HH	Hamburg	040- 5381657
K	Köln	0221- 1390008
KI	Kiel	0431- 6498116
KR (QUARK)	Krefeld	02151- 307007
M	Muenchen	089- 2605293
M2	Muenchen	089- 5807938
(M3 (infiMaus)	Muenchen	089- 659769 off?)
MK	Iserlohn	02371- 14490
N	Nürnberg	0911- 405167
RD	Rendsburg	04331- 80866
S	Stuttgart	0711- 264791
S2	Stuttgart	0711- 6364625
TL	Bad Tölz	08041- 73262
W	Würzburg	0931- 18993

FidoNet, Area Forth.Ger

engine&arts Hoenigtal



Forth-Gruppen regional

W-1000 Berlin	Claus Vogt Tel. 0+30 - 2 16 89 38 p Treffen: nach Absprache
W-4XXX Rhein-Ruhr	Jörg Plewe Tel. 0+208 - 42 35 14 p Treffen: jeden 1. Samstag im Monat im S-Bahnhof Derendorf Münsterstr. 199, 4000 Düsseldorf
W-4130 Moers	Friedrich Prinz Tel. 0+2841 - 5 83 98 p Treffen: jeden Samstag 14:00 Arbeitslosenzentrum, Donaustr. 1 4130 Moers
W-51XX Aachen	Arndt Klingelberg, Tel. 0+2404 - 6 16 48 agp Treffen: jeden 1. Montag im Monat als Gruppe des Computer-Club der RWTH, Seminargebäude Raum 214, Nähe Wüllnerstraße 5100 Aachen
W-6800 Mannheim	Thomas Prinz Tel. 0+6271 - 28 30 p Ewald Rieger Tel. 0+6239 - 86 32 p Treffen: jeden 1. Mittwoch im Mo- nat, Vereinslokal Segelverein Mannheim e.V., Flugplatz 6800 Mannheim-Neuostheim

Forth-Fachgruppen

z. Zt. keine Gruppe

Gruppengründungen, Kontakte

Regional

W-7000 Stuttgart Wolf-Helge Neumann
Tel. 0+711- 8 87 26 38 p

Fachbezogen

8051 ... (Forth statt Basic, e-FORTH)
Thomas Prinz
Tel. 0+6271 - 28 30 p

Forth-Hilfe für Ratsuchende:

Forth allgemein

Jörg Plewe
E-Mail-Adresse:
plewe@mpi-dortmund.mpg.de
Tel. 0+208 - 42 35 14 p
Karl Schroer
Tel. 0+2845 - 2 89 51 p
Jörg Staben
Tel. 0+2103 - 24 06 09 p
Di. und Fr.
Frank Stüss
Tel. 0+6187 - 9 15 03 ap
Gerd Limbach
Tel. 0+2051 - 25 51 12 p
Mo. + Di. 20:00 - 22:00

**Anfänger und
Wiedereinsteiger**

Spezielle Fachgebiete

32FORTH (Atari)	Rainer Aumiller Tel. 0+89 - 6 70 83 55 gp
FORTHchips (FRP1600, RTX, Novix ...)	Klaus Schleisiek-Kern Tel. 0+40 - 2 20 25 39 gp
F-PC & tCOM, ASYST (Meßtechnik), embedded controller (H8/5xx//TDS2020, 8051 ... eFORTH...)	Arndt Klingelberg Tel. 0+2404 - 6 16 48 agp
Gleitkomma-Arithmetik	Andreas Döring Tel. 0+721- 59 39 35 p
HS/Forth (Harvard Softworks)	Wigand Gawenda Tel. 0+30- 44 69 41 p
KI (Künstliche Intelligenz), OOF (Object Oriented Forth)	Ulrich Hoffmann Tel. 0+431 - 80 12 14 p
Unterricht mit FORTH	Rolf Kretzschmar Tel./Fax 0+2401 - 8 88 91 ap
UUCP (FORTH ... per eMAIL)	Andreas Jennen Tel. 0+30 - 3 96 52 27 ap
volksFORTH/ultraFORTH	Klaus Kohl, Tel. 0+8233 - 3 05 24 p

Forth-Forum: Forth-Mailbox

Jens Wilke (SysOp)
Tel. 0+89 - 8 71 43 52 p
Mailbox 0+89 - 8 71 45 48,
300-2400 baud (8N1)

Hinweise

Zu den Telefonnummern

a == Anruferbeantworter, 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

Für unsere Leser aus den neuen Bundesländern sind die Ortsnetzkennzahlen auch nach 91jun01 nicht eindeutig, bitte informieren Sie sich über örtliche Besonderheiten.

Appell

Es werden weitere Kontakte / Ansprechpartner gesucht, insbesondere aber auch im Osten. Bitte wenden Sie sich an die Redaktion (A. Klingelberg, Straßburger Straße 12, W-5110 Alsdorf, Tel 0 24 04-6 16 48, Fax 0 24 04-6 30 39).

Redaktionsadresse

Rolf Kretzschmar, Rote Gasse 7, W-5112 Baesweiler
Tel./Fax 0 24 01-8 88 91

Als international tätiger EDV-Hersteller mit extremen Zuwachsraten suchen wir einen

Entwicklungsingenieur für hardwarenahe Softwareentwicklung

Sie sollten als Diplom-Ingenieur oder Physiker neben den fachlichen Fähigkeiten (Assembler, höhere Programmiersprachen, Betriebssysteme) über eine ausgeprägte Kooperationsbereitschaft, die selbständiges Arbeiten jedoch einschließt, und gute Englischkenntnisse verfügen.

Nun ist die Reihe an Ihnen. Wenn Sie bei uns starten wollen, sollten Sie uns Ihre Bewerbungsunterlagen zusenden. Wir möchten mehr über Sie erfahren und Ihnen mitteilen, was GFC für Sie in Zukunft bereit hält.



mbH, Vogelsanger Weg 39, 4000 Düsseldorf 30
Telefon (0211) 6 1895-0, Telefax (0211) 6 1895-74

Hamburg · Bielefeld · Halle · **Düsseldorf** · Frankfurt · Stuttgart · München

KK-FORTH hatein Herz für Mikrocontroller

- Verfügbar für:
- IBM-PC/XT/AT oder Kompatible
 - mc-PC-EMUF (V20)
 - mc-Z80-Mini-EMUF (84C015)
 - mc-RISC-EMUF oder FG-RTX-Board (RTX2000/1A)
- Benötigt:
- mindestens 16KByte EPROM; ab 8KByte RAM (optional: 32K/32K)
 - eine serielle Schnittstelle (PC-Terminalprogramm wird mitgeliefert)
- Features:
- über 400 Befehle (FORTH-83 mit einigen ANSI-Erweiterungen)
 - ROM-Fähig, für Interrupts vorbereitet
 - Komfortabler Zeileneditor, Fileinterface über PC-Terminalprogramm
 - Viele Kernroutinen umleitbar (z.B. Ein-/Ausgabe)
- Preise (inkl. MwSt):
- | | |
|-----------|--|
| DM 65.00 | Handbuch allein (für alle Versionen gleich) |
| DM 70.00 | Eine Version (Disketten und Zusatzbeschreibung) |
| DM 110.00 | Vollversion (Disketten und vollständiges Handbuch) |
- Zilog Super8-Chip mit ROM-FORTH
- | | |
|-----------|---|
| DM 45.60 | Super8-Chip mit FORTH im Masken-ROM |
| DM 34.20 | Beschreibung und Disketten zum S8-FORTH |
| DM 51.30 | Bausatz zum S8-Projekt VD3/91 (Platine, GAL, SMD's) |
| DM 228.00 | Vollständig bestückte Platine mit Beschreibungen |

Ingenieurbüro Klaus Kohl · Pestalozzistraße 69 · W-8905 Mering 1



FORTH-SYSTEME GMBH

Postfach 1103,
7814 Breisach

Telefon (0 76 67) 5 51
Telefax (0 76 67) 5 55

Telefon Schweiz:
(055) 53 65 55

UR/FORTH

- FORTH-83 Standard
- Für MS-DOS, OS/2, 80386
- Direkt gefädelt Code Implementationen mit dem obersten Stackwert im Register um größtmögliche Ausführungsgeschwindigkeit zu erreichen
- Segmentiertes Speichermodell mit Programm, Daten, Headers und Dictionary Hash Table jeweils in einem getrennten Segment
- Komplett gehashtes Dictionary führt zu extrem schneller Übersetzung
- Mächtige neue String Operatoren (Suche, Extraktion, Vergleich und Addition) sowie einen dynamischen String-, Speichermanager
- Kann mit Objektmodulen, die in Assembler oder anderen Hochsprachen erzeugt wurden, gelinkt werden
- Native Code Optimizer zur direkten Umsetzung in 80 x 86 Code im Lieferumfang

WinFORTH

- UR/FORTH kompatibel
- Windows Funktionen werden voll unterstützt
- Erweiterte Debugging-Hilfsmittel
- Online Windows Hilfe
- Coprozessor Unterstützung möglich
- Software-Gleitkomma-Paket
- Viele Beispielprogramme
- Upgrades von UR/FORTH Systemen auf WinFORTH sind preisgünstig zu erhalten

SRS II

- Serieller ROM Emulator
- Unterstützung folgender Bausteine:
27256, 27512, 271000, 27010, 27020, 27040
- Minimale Zugriffszeit 100 ns
- Maximale Baudrate 115.200 bits/s
- Highspeed Interface als Option
- Gleichzeitiger Zugriff von Host und Zielprozessor
- Zusätzliche serielle Schnittstelle über den ROM-Sockel
- Intel-Hex, Motorola-S oder ASCII/binär Formate werden unterstützt
- Der SRS II ist nur 157 x 94 x 36 mm groß
- SRS63 kompatibel

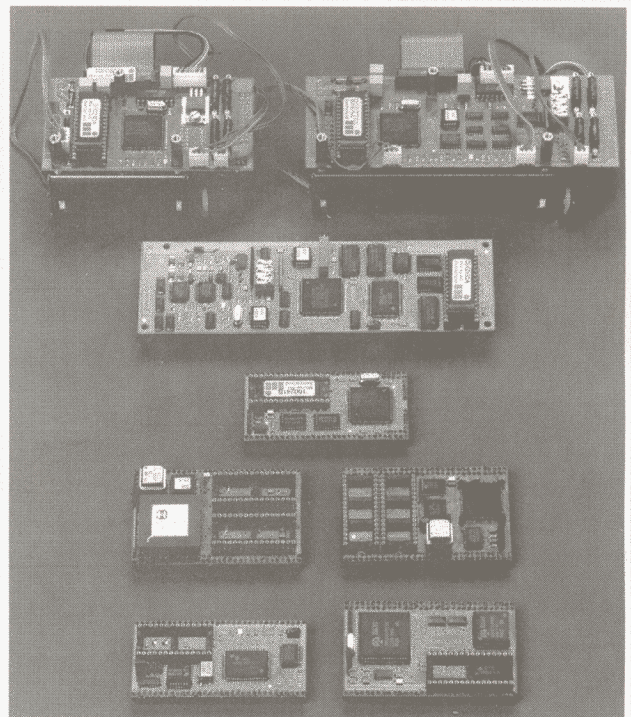
FORTH-83 Metacompiler

Der LMI FORTH Metacompiler wird mit komplettem Quellcode für ein ausführlich ausgetestetes, Hochgeschwindigkeits FORTH 83 Kern ausgeliefert, wobei Sie die Auswahl aus folgenden Zielprozessoren haben:

- | | |
|---------------|---------------|
| • 8086/8088 | • 78310 |
| • Z80/HD64180 | • 8031/32/535 |
| • 8080/8085 | • 6303 |
| • 68000 | • 6502 |
| • Z8 | • V25 |
| • 1802 | • 68HC11 |
| • 6809 | • RTX 2000 |
| • 8096/97 | • 80C166 |

Sie erzeugen schnelle und kompakte Anwendungen, indem Sie Ihre Quellprogramme mit unserem Forth Nucleus zusammenstellen und ihn mit dem LMI FORTH Metacompiler übersetzen.

ModuNORM



CPU-Steck-Module im Scheckkartenformat:

- 8 Bit z.B. 6303
- 16 Bit z.B. V25
- Highspeed RTX-2000/1
- 80C166
- Softwareunterstützung durch SwissFORTH
- Thermodrucker und Controller
- LCD Grafik-Controller

Bitte fordern Sie unseren Produktkatalog und die Preisliste an. FORTH-Gesellschaftsmitglieder erhalten bis zu 10% Rabatt (artikelabhängig).