

*BASIS15*

*Zeitbezogenen booten*

*Neues im Core Word Set*

*Dynamischer Stringbuffer*

*EuroFORML'91*

**FORTH  
MAGAZIN**

## Hier könnte Ihre Anzeige stehen:

Gültige Anzeigenpreise und -formate bei:

FORTH-Gesellschaft e.V., W-8044 Unterschleißheim, Postfach 1110,

Tel. 089-317 37 84

oder

DFÜ 089-871 45 48 secretary

Postgiroamt 2000 Hamburg, Kontonummer: 56 32 11-208

Bankleitzahl : 200 100 20

und

Redaktion:

Peter Dinies, Metzstraße 38, W-2300 Kiel 1, Tel. 04 31-1 32 39

## **EDV-Beratung - Software-Design - Goppold**

*Bgm. Germeierstraße 4 - 8011 Poing - Tel.: 0 81 21 - 8 27 10*

### Wir haben das FORTH Know-How

- > Consulting, Projekt-Management, Beratung, Schulung.
- > FORTH unter UNIX (alle Systeme), SUN-Workstations (SPARC&68k), PC-Systeme, FORTH-Prozessoren.
- > Eigen-Entwicklung fortgeschrittener Software-Technologie: Objekt-Programmierung, Datenbanken, Hypertext
- > Distributor für MPE: Single Chip Targets&Boards, RTX 2000, Eprom-Emulatoren, Modular FORTH, Power FORTH (Katalog DM 5,- im Vorverkauf oder Nachnahme). Desweiteren Vermittlung von: LMI, Harvard Softworks
- > Und natürlich Leibniz, das System nach FORTH

Demo-Disk: DM 20,- VK oder NN

# FORTH-Magazin

## Vierte Dimension

Nr. 2 Juni 1991

- 4 20 Jahre Beschleuniger (Peter Dinies)
- 5 Impressum, Editorial
- 6 Nachrichten
- 7 FORTH-Tagung in München (Michael Kalus)
- 8 Besprechung des ANS-FORTH (Michael Kalus)
- 15 Der dynamische Stringbuffer (Heinrich Hohl)
- 16 FORTH ganz praktisch (Friedrich Prinz)
- 24 Problembeschreibung: Cursor (Jörg Staben)
- 25 Neues im CORE WORD SET (Michael Kalus)
- 31 Die neue Münchner FORTH-Mailbox (Jens Wilke)
- 37 A taste of.... (Ulrich Hoffmann)
- 38 7th euroFORML conference (Klaus Schleisiek-Kern)
- 39 FORTH-Gruppen
- Anzeigen:
- |  |       |
|--|-------|
| EDV-Beratung - Software-Design, Goppold, Poing | S. 2  |
| FORTech Software, Rostock                      | S. 3  |
| FFORTH für Atari ST, Galactic, Essen           | S. 14 |
| big-FORTH, Bernd Paysan, München               | S. 27 |
| FORTH-Systeme, Angelika Flesch, Mikrap AG      | S. 36 |
- Kleinanzeigen S. 27

# FORTech

Software

**Wir haben ihn gewonnen,**

... den Programmierwettbewerb zur Messe ECHTZEIT '90.

Und von den Kunden, für die wir seit fünf Jahren mit dem Siegersystem comFORTH arbeiten, hat sich (natürlich) keiner gewundert.

**Auch für Sie zu haben:**  
**das Originalsystem comFORTH**

wahlweise z. B. für:

- Totalkontrolle über Ihren PC
- Firmware-Entwicklung
- Echtzeitanwendungen in der Automation
- Programmierung verteilter Rechnersysteme
- KI-Probleme
- Numerische Aufgaben

Das ist akkumulierte akademische Brainpower - nicht von Informatikern, sondern von Ingenieuren für Ingenieure plus zehn Jahre praktische Automation potenziert mit zehn Jahren Forth-Know-how.

**Automatisierung -  
unsere Spezialstrecke:  
Wir automatisieren a l l e s.**

**FORTech Software GmbH**  
**Albert-Einstein-Straße 2**  
**O 2500 Rostock 6**  
**Telefon 40 55 96**  
oder: **c/o Becker & Partner**  
**Bremer Straße 18**  
**W 2100 Hamburg 90**

### Gratis-Coupon

**Mich interessiert:**

- comFORTH - Infos
- comFORTH - Preisliste
- Demo - Disk comFORTH
- Automatisierung

**Mein Automatisierungsvorhaben:**

\_\_\_\_\_

Mein Name/Firma: \_\_\_\_\_

Straße/Nr.: \_\_\_\_\_

PLZ/Ort: \_\_\_\_\_

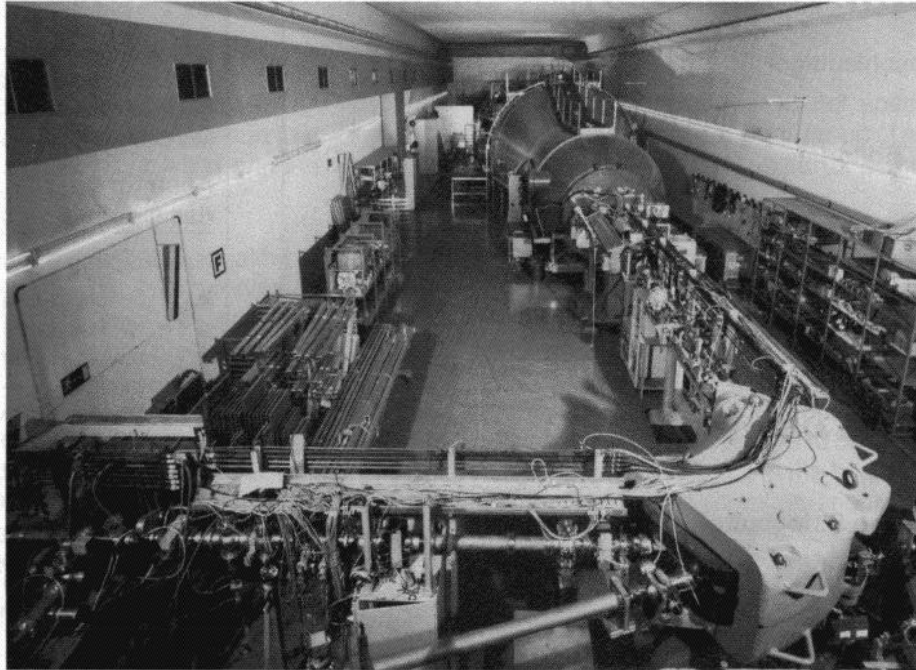
## Zwanzig Jahre Beschleuniger

Anläßlich der Jahrestagung FORTH '91 und des bevorstehenden Jubiläums "Zwanzig Jahre Münchner Tandem-Beschleuniger" in Garching, besuchten die Tagungsteilnehmer das Beschleunigerlaboratorium der

die Spannungen bis 15 Millionen Volt erlauben werden.

Etwa ein Dutzend Meßeinrichtungen stehen dem Institut zur Verfügung, darunter ein Spektrometer, das Präzisionsmessungen von Kernmas-

Grundwasser in einer wasserführenden Sandsteinschicht nach eine Fließstrecke von 100 Kilometer zu einer Million Jahre gemessen. Das entspricht einer Fließgeschwindigkeit von 1 Meter in 10 Jahren. Zum



Ludwig-Maximilians-Universität und der Technischen Universität München.

Wie unser Ex-Direktor der FORTH-Gesellschaft e.V. Heinz Schnitter (Heinz Schnitter hat sich auch in bemerkenswerter Weise dafür eingesetzt, daß die Steuerung der Anlage durch die Programmiersprache FORTH erfolgt) uns mitteilte werden Beschleuniger benötigt, um Energieschwellen zwischen Atomkernen oder zwischen Atomen überwinden zu können. Solche Schwellen sind z.B. die elektrische Abstoßung zweier Atomkerne oder das Abstreifen aller Elektronen eines Ions beim Durchgang durch dünne Schichten. Im ersten Fall werden sie als Mikroskope zur Untersuchung der Atomkerne eingesetzt.

Weiter führte unser EX aus: die Hauptmaschine des Beschleunigerlabors ist ein Tandem-van-de-Graaff-Beschleuniger mit einer Gleichspannung von gut 12 Millionen Volt. Derzeit wird die Maschine mit neuen Beschleunigungsröhren ausgerüstet,

sen und Kernenergien erlaubt, und zwei im Haus gebaute Hochfrequenz-Nachbeschleuniger mit einer totalen Spannung von 10 Millionen Volt. Ein supraleitender Nachbeschleuniger nach dem Zyklotron-Prinzip steht kurz vor der Fertigstellung.

Die technologischen Schwerpunkte im Garchinger Labor im Rahmen zahlreicher Diplom- und Doktorarbeiten sind: die Entwicklung von neuartigen Teilchenbeschleunigern, in letzter Zeit unter Nutzung der Supraleitung, von neuartigen Teilchendetektoren, effizienten Ionenquellen und die Herstellung dünner Schichten höchster Homogenität.

Die heutigen Hauptaktivitäten sind Massenspektrometrie mit Beschleunigern, Materialanalysen mit Ionenstrahlen, Präzisionsmessungen von Kernmassen und -energien sowie vollständige Beschreibungen von Kernanregungen. So kann z.B. ein Chlor-36-Atom unter einer Million Milliarde Chlor-Atomen nachgewiesen werden. So wurde das Alter von

Vergleich: vom GSF-Institut für Hydrologie abgeschätzt, ist das Münchener Brauereiwasser etwa 10.000 bis 14.000 Jahre alt. Andere Objekte sind die Bestimmung von Erosionsraten über die Messung der Konzentration von Radioisotopen, die über Höhenstrahlung in der Erdkruste erzeugt wurden, oder Tests von fundamentalen Prinzipien in der Physik, wie dem Pauliprinzip, das es verbietet, daß mehr als ein Elektron oder Proton einen quantenmechanischen Zustand besetzt.

Finanziert wurden die Arbeiten am Beschleunigerlaboratorium der Münchener Universitäten durch den Freistaat Bayern. Zusätzlich hat der BMFT, im Rahmen der Verbundforschung, und der BMW, im Rahmen der Hochschulbauförderung, viele der Arbeiten in München und an den auswärtigen Großforschungszentren mit Personal- und Sachmitteln unterstützt.

**Peter Dinies**

## FORTH-Magazin "Vierte Dimension"

### Herausgeber:

FORTH-Gesellschaft e.V.

### Redaktion, Satz und Layout:

Peter Dinies, Metzstraße 38, W-2300 Kiel 1,  
Tel. 0431/1 32 39

### Redaktioneller Beirat:

Ulrich Hoffmann, Jens Storjohann, Michael Kalus,  
Rolf Kretzschmar

### Kontaktadresse:

FORTH-Büro, Postfach 1110, W-8044 Unterschleiß-  
heim, Tel. 0 89/3 17 37 84 oder FORTH-Mailbox,  
München, Tel. 0 89/7 25 96 25 8 N1  
"Konferenz Vierte Dimension".

### Quelltextservice:

Der Quelltext von Beiträgen, die mit dem Disketten-  
symbol gekennzeichnet sind, ist auf der Leserservi-  
ce-Diskette zur jeweiligen Ausgabe oder in der  
FORTH-Mailbox zu finden.

### Autoren dieser Ausgabe:

Michael Kalus, Heinrich Hohl, Jörg Staben,  
Friedrich Prinz, Jens Wilke

### Redaktionsschluß:

Erste Woche im mittleren Quartalsmonat  
Erscheinungsweise vierteljährlich

### Auflage:

Ca. 1.000

### Druck:

Buch- und Offsetdruckerei Bickel & Söhne, Frankfur-  
ter Ring 243, W-8000 München 40

### Preis:

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

### Rechte:

Berücksichtigt werden alle eingesandten Manuskrip-  
te von Mitgliedern und Nichtmitgliedern.  
Für die mit Namen oder Signatur des Verfassers ge-  
kennzeichneten Beiträge übernimmt die Redaktion  
lediglich die presserechtliche Verantwortung. Die in  
diesem Magazin veröffentlichten Beiträge sind urber-  
rechtlich geschützt. Übersetzung, Vervielfältigung,  
Nachdruck sowie Speicherung auf beliebige Medien  
ist auszugsweise mit genauer Quellenangabe er-  
laubt. Die Beiträge müssen frei von Ansprüchen Drit-  
ter sein. Veröffentlichte Programme gehen, soweit  
nicht anders vermerkt, in die Public Domain über.  
Für Fehler im Text, in Schaltbildern, Aufbauski-  
zen etc., die zum Nichtfunktionieren oder evtl. Schadhaf-  
werden von Bauelementen führen, kann keine Haf-  
tung übernommen werden. Sämtliche Veröffentli-  
chungen erfolgen ohne Berücksichtigung eines  
eventuellen Patentschutzes, auch werden Warenna-  
men ohne Gewährleistung einer freien Verwendung  
benutzt.

Das interessanteste für mich in der letzten Zeit bezüglich  
Forth war ANS Forth. Das Dokument dazu heißt BASIS15.  
Die Welt der PC's hat dazu schon ein Forth in Händen: FPC  
von Tom Zimmer wird wohl zum Modell erhoben und den  
ANS-Aufbau bekommen. Aber bitte keinen Schreck bekom-  
men - es geht auch kleiner. Unter dem Dach des ANS haben  
viele Platz gefunden. Wird der Standard die weitere Verbrei-  
tung von FORTH fördern?

Betrüblich dagegen, daß Forth in Silicium bisher keine  
Kunden finden konnte. Eine Zeitlang herrschte reges Interes-  
se an diesen Chips, jetzt ist es ruhiger geworden um diese  
Hochsprachen-Prozessoren. War alles nur eine Spielerei,  
oder ist dies jetzt die Ruhe vor dem Sturm? Es gibt sie ja nach  
wie vor - und neue kommen gerade dazu: Super8 FORTH im  
8Kbyte On-Chip ROM und ein Forth-RISC-Processor FRP  
1600. Gesucht: Massenanwendungen. Sachdienliche Hin-  
weise an die Redaktion.

Der dynamische Stringbuffer des LMI-Forth gefällt. Ich bin  
sicher, es gibt noch sehr viel berichtenswertes dieser Art.  
Was 'praktisch' ist muß sicherlich jeder für sich entscheiden.  
Anwendungen aber, geschrieben in FORTH, die für jeman-  
den praktisch waren, sehe ich gerne. Besten Dank also dafür.

Nachzutragen wären dann noch einige Korrekturen zum  
letzten Heft - besten Dank übrigens den aufmerksamen Le-  
sers für die Hinweise. Die Telefonnummer in eigenen Anzei-  
ge der FORTH-Gesellschaft hatte eine 7 zuwenig - aber im  
Impressum war sie korrekt. Einige "Drückfehlerteufel" blie-  
ben unentdeckt, haben aber keinen größeren Schaden ange-  
richtet - hoffen wir mal. Die beiden Autoren Jörg Plewe und  
Jörg Staben bitten wir um Entschuldigung, weil der Hinweis  
fehlte, daß die Quelltexte zu ihren Beiträgen "Universelles  
Filterwerkzeug" und "Die unendliche Geschichte oder Lokale  
Variablen" nicht abgedruckt wurden - zu lang. Die gibt es auf  
Anfrage als Diskette. Wo? Bei den Autoren sicherlich - ich  
denke sie werden sich über Resonanz freuen - und beim  
Quellcodeservice der FG [ ??? der sätza ]

Die Anregungen und die Kritik, die die Redaktion auf  
unserem Treffen in München bekommen hat, waren uns  
nützlich und haben die 'Linie', die wir fahren wollen beein-  
flußt. Wie? Das wollen wir mit den folgenden Heften zeigen  
und hoffen weiterhin auf lebhaft Kritik.

Herzlichst  
Eure Redaktion

P.S.

Die euroFORMLconference ist dieses Jahr in der Tsche-  
choslowakei. Sie findet vom 11. bis 13. Oktober statt. Es ist  
übrigens die 7te europäische euroFORML.

**Originalautor** :POSTMASTER\_FORTHEV@INFINET.ZER  
**Originalbetreff** :1737 ANS FORTH TECHNICAL COMMITTEE

**Path**: forthev!unido!mcsun!many.more!unix.cis.pitt.edu!pitt!willett!ForthNet  
**From**: ForthNet@willett.pgh.pa.us (ForthNet articles from GENie)  
**Newsgroups**: comp.lang.forth  
**Subject**: ANS FORTH TECHNICAL COMMITTEE  
**Message-ID**: 1UUL1.3#5129@willett.pgh.pa.us  
**Organization**: (n.) to be organized. But that's not important right now.

D.RUFFER [Dennis] at 20:21 EDT

TO: X3J14-watchers

FROM: Elizabeth D. Rather, Chair

I'm writing to keep you up to date on the status of the developing standard.

In April, the TC voted 17-2 to submit our current draft as a dpANS. Exactly what this means will be discussed shortly. May 3-4 we met to consider the two negative votes. During this process, it became clear that additional "clean up" is necessary; to that end we have scheduled another meeting for June 16-18 at the University of Rochester, Rochester, NY (immediately preceding the Rochester Conference). The time and place was chosen to make it as easy as possible for those of us attending the Rochester Conference; if any of you who are interested to join us, you'll be welcome. Contact the Forth Institute or me if you need further information. At this time, primary attention is focussed on making the document as clear and consistent as possible. For example, much work is currently being done on organization, rationales, additional material to help people understand differences between ANS Forth and FORTH-83. A copy of BASIS16 (resulting from the May meeting) will be available about June 1.

At the end of the June meeting, we will vote in the TC on submitting the results of that meeting to a letter ballot to approve release as a dpANS. If there are any negative votes on that letter ballot, they will be considered at our next meeting beginning July 30 in Boulder, CO. In any case, it is likely that a dpANS will go out for public review in approximately September.

What does this mean?

Few standards are as publicly distributed during the development process as this one. In most cases, the first widespread public dissemination of a proposed standard comes with the first 4-month public review. The objective is to get as broad a distribution as possible, and solicit as much public comment as possible. There will be a press release issued by ANSI, and every effort will be made to tell people that it's available for review and how to get it.

The document will be sold through an organization called Global Engineering Documents. They not only sell draft and final standards, they also maintain a mailing list of people who buy copies. The reason for this is that when subsequent drafts (and ultimately the final ANS) are published, Global will automatically contact everyone who bought the last one and notify them that a new one is available. We don't know how much it will be; documents of similar size tend to run around \$60 (I'm told people have complained about these prices for years -- Global swears it's near their cost. Remember too that you're paying to be on the official list). The dpANS will \*not\* be available from FORTH, Inc., on a BBS, or anywhere but Global. The purpose of this is to ensure that the proper tracking is done (people have been known to sue because nobody told them that a subsequent draft standard had been released).

Comments can be positive or negative. They should be directed to ANSI, who will not only route them to us but will monitor our responses to all negative votes to ensure they're "responsive". The value of these comments, to the commenter as well as to us, will be directly proportional to their specificity. At our May meeting we had a guest from X3 who told us that someone wrote another TC, "your standard is so bad it's only fit for the fireplace." The TC responded, "we're glad you'll be warm this winter." On the other hand, if you call our attention to some critical technical flaw, a misstatement in a rationale, or some important piece of information missing or hard to find, for example, we will make every effort to fix it. We'll also carefully consider requests for missing features or to remove unpopular features, although in these cases we'll be looking more for widespread demand and are less likely to make a change in response to an individual request.

ANSI will follow up as to whether you've found our response acceptable (a "resolved negative") or not (an "unresolved negative"). Unresolved negatives follow the document around until it is a final ANS. If ANSI thinks there are too many of them, or that our responses are inadequate, they can hold up further processing.

We (the TC) can make any changes at any time. There's been some talk to the effect that at some point it's "out of our hands." This is never true until the final publication of the ANS. I've heard of TCs that stopped a

standard days before its final publication because they wanted to change something. In theory, changes get harder to make, because a 2/3 vote is required. But in fact, operating rules require "consensus," which means in practice that virtually all actions are unanimous, and the rest have very small minorities (under 3 votes). Since our decisions have been reached by consensus, frequently achieved only after many hours of debate over several meetings, compromises, revisions, etc., we are unlikely to change them lightly. We were very open to public input throughout our process; we received lots, and it was very influential. At this stage, however, most issues have been dealt with at length and to our technical satisfaction.

It is true that if any changes are made the document must repeat the cycle of letter ballot, approval by X3, and public review. After the first 4-mo. review, subsequent review periods are 2-mos.

We're expecting the period during and after the public reviews to be active, busy and important. We're looking forward to getting some broad public response to our work. We also think it's very likely that there will be at least one more review period, because of changes either in response to public comments or internally generated.

When the happy day arrives when there really is an ANS Forth, its legal status is as follows: no one is required to build systems that conform to it, and no one is required to write ANS Forth programs (at least, not required by ANSI.

Your boss may...). Just like previous Forth standards, it will influence public usage precisely to the extent it is voluntarily adopted by implementors and demanded by users. **\*\*HOWEVER\*\*** -- UNLIKE FORTH-83, if someone claims to have produced an ANS Forth system, and uses that labeling, but is in violation of the standard, ANSI can force the labeling to be removed. They would do so in response to public complaints -- they haven't the capacity to "vet" all systems produced.

I hope this clarifies things for you. We look forward to hearing from you.

-----

This message came from GENie via willett. You \*cannot\* reply to the author using e-mail. Please post a follow-up article, or use any instructions the author may have included (USMail addresses, telephone #, etc.). Report problems to: [dwp@willett.pgh.pa.us](mailto:dwp@willett.pgh.pa.us) \_or\_ [uunet!willett!dwp](mailto:uunet!willett!dwp)

---

## FORTH '91

### Jahrestagung in München

*Freitag, 12. April, strahlender Sonnenschein in der bayrischen Metropole. Es war eine bequeme Anreise per Nachtzug. Mit dem Taxi ging es hinaus zur Schießanlage nach Hochbrück. Erinnerungen an die Olympiade. Der Anlage war die einstige Pracht inzwischen etwas abhanden gekommen, Wegplatten lagen schief, Gras wächst in den Ritzen, die Farbe der Gebäude verblasst zusehens. Großzügig waren die Planer damals, viel Raum beeindruckt den Besucher nach wie vor.*

*Unsere Jahrestagung der Forthgesellschaft war jetzt wieder in München. Eigentlich sollte das Treffen ja in der Umgebung von Rostock oder Dresden sein, jetzt, wo doch die neuen Bundesländer dabei sind. So war es beschlossen worden im letzten*

*Jahr in Frankfurt/Main. Doch die Wende dort erlaubte keine verbindliche Planung dort, man sah sich außerstande so ein Treffen abzuhalten. München kam zur Hilfe.*

*Im Hotel Olympia bot man uns eine vorzügliche Unterbringung und gute Küche, freundlichen Service und Hilfsbereitschaft bei der Tagung. Die entspannte Atmosphäre ist mir sehr angenehm und wohltuend in Erinnerung geblieben - den meisten ging es wohl so.*

*Es war alles gut vorbereitet - Anmeldung, Tagungsmaterial, Tagungsprogramm. Am Freitag Nachmittag bald nach 14 Uhr eröffnete Heinz Schnitter als unser Gastgeber die Tagung. Vorbereitet waren neun verschiedene Workshops, aber die*

*Teilnehmer wählten daraus zwei Beiträge aus, die dann vor der Versammlung vorgetragen wurden: ANS Forth BASIS15 und eine Musikvorführung per Forth auf Midi-Maschinen der Gruppe CAMP Berlin welche mir außerordentlich gefallen hat. Später fuhren wir zur Besichtigung des Beschleunigerlaboratoriums der TU und LMU München, um dort die Anlage selbst einmal anzusehen und deren Steuerung zu erleben, welche natürlich auch mit Forth realisiert worden ist. Bis zum Abendessen blieb noch genügend Zeit für allerlei Gespräche am Rande und Fachsimpelerei. Der gesamte Rahmen war erfreulich locker und wenig bepackt mit Terminen.*

**Fortsetzung auf Seite 30**

# BASIS15

## Besprechung des ANS-Forth Entwurfs - Stand 4/91

*Die Anstrengungen, einen amerikanischen nationalen Standard der Programmiersprache FORTH (ANS-FORTH) zu erarbeiten, werden wohl erfolgreich sein. Das technische Komitee (TC) für Forth Programmiersysteme ANS X3J14 dokumentiert mit BASIS15 den Stand der Arbeit. Dieses Dokument soll Entwurf für den ANS-Forth Antrag sein (draft proposed ANS = dpANS). Die Beratungen gingen von November 1987 bis Februar 1991 in bisher 14 Sitzungen und dazwischen ausgiebiger Korrespondenz. Das Ergebnis ist eine sorgfältige Zusammenstellung all dessen, was seit der Erfindung von Forth durch Charles Moore als brauchbarer Bestandteil dieser Programmiersprache erkannt worden ist und für unverzichtbar bis sehr empfehlenswert gehalten wird. BASIS15 beschreibt den derzeitigen soliden Wortschatz von Forth in allen Einzelheiten (word sets). Zuvor wird das Konzept der Sprache erklärt. Im Anhang geht man ausführlich auf die Entwicklung einzelner Forthworte und Konzepte ein.*

Mit der auszugsweisen Besprechung möchte ich auf die herausragende Bedeutung dieses Dokumentes hinweisen - ein Muß für jeden, der in FORTH programmiert.

### **Zur Geschichte des BASIS15**

Mr. Charles Moore erfand Forth, um die Produktivität von Programmierern zu steigern, ohne dabei die Effizienz der zugrunde liegenden Maschine zu verlieren. Forth wurde eine Programmierumgebung mit mehreren Schichten. Neben Elementen einer Computersprache enthält es auch solche des Betriebs-Systems und einen Maschinen-Monitor. Diese erweiterbare Umgebung in verschiedenen Ebenen erlaubt ein äußerst interaktives Programmieren und Testen von Entwicklungen (1960-1970, 1971 erste komplette Implementation für das 11-Meter Radioteleskop am National Radio Astronomy Observatory, Arizona).

Um Applikationen besser portieren zu können, kümmerten sich Mitte der 70er Jahre europäische Forth Benutzer um mehr Einheitlichkeit im Forth. Im Jahre 1977 formulierten sie in Utrecht den FORTH-77 Standard. Im gleichen Jahr brachte die FORTH,

Inc. - noch mit C. Moore - das micro-FORTH heraus. 1978 wurde in den USA die Forth Interest Group (FIG) gegründet. Sie verbreitete FIG-Forth, eine abgespeckte public domain Version des Forth aus dem Observatorium.

Die Erfahrungen mit Forth wurden breiter. Das Forth Standards Team entstand und veröffentlichte die weiteren Entwicklungen als FORTH-78 und FORTH-79. Die verschiedenen Produkte entwickelten sich parallel. Man entdeckte mehrere unschöne Festlegungen in den "Standards". Schließlich einigte man sich 1983 auf eine überarbeitete Fassung. FORTH-83 blieb dann relativ stabil und viele Anbieter erfüllten den Standard. In der Zwischenzeit reifte aber die Überlegung, daß man einen autorisierten Standard benötige. Das technische Komitee für Forth Programmiersysteme wurde im November 1987 in den USA gegründet und hat nun nach drei Jahren der Bestandsaufnahme den Vorschlag zur Anerkennung als ANS-FORTH verabschiedet. Daraus soll noch in diesem Jahr das dpANS-FORTH werden, welches dann ein Jahr lang der öffentlichen Kritik ausgesetzt sein

wird. Dann endlich wird es wohl den Standard geben: ANS-FORTH.

### **Das Konzept von ANS-FORTH**

Das Forth-Konzept des Standards wurde auf die wesentlichen Elemente zurück geführt.

Forth ist ungewöhnlich, weil es mehr ein semantisches Modell als ein syntaktisches ist. Das Symbol trägt die Bedeutung und nicht so sehr die Form des Ausdrucks. Ein in ANS-Forth definiertes Symbol ist ausführbar, und in der Regel ist der operationale Effekt gleich mit der Bedeutung des Symbols. Jedes Symbol hat eine Aktion und die Aktion ist seine Bedeutung. Die Bedeutung eines Forthprogramms ergibt sich aus der Bedeutung der Phrasen des Konstruktes.

Eine Applikation bildet die ihr innewohnende natürliche Grammatik ab. Das ANS-Forth selbst hat nur wenig bis gar keine eigene Grammatik, es ist erweiterbar und kann jede sachgerechte Grammatik abbilden. Charles Moore meinte, Forth verschmelze die zweckdienliche Sprache einer Applikation mit der Computersprache, so daß daraus eins wird.



Wie jede Computersprache schließt Forth die semantische Lücke zwischen der Hardware und den Menschen. Während die menschliche Kommunikation Umgangssprache (natural language) verwendet, reagieren Computer nur auf Sequenzen von Bits. Forth übersetzt menschliche Anweisungen auf das Niveau der Computerhardware, indem es geeignete Bit-Sequenzen in Worte packt und es möglich macht, Worte höherer Ordnung aus primitiveren Worten zu bilden, bis die nötige Funktionalität erreicht ist. Die Definition eines Wortes ist einfach eine Zusammenfassung anderer Worte, die wiederum aus anderen Worten bestehen usw. Das fundamentalste Element in einem Forthprogramm ist also das Wort.

#### **Das Wörterbuch**

Forth organisiert diese Worte in Strukturen, die durchsucht werden können, und nennt diese Wörterbuch (dictionary). Man nennt es compilieren, wenn der Quellcode für ein neues Wort in eine ausführbare Form gebracht, und dann in das Wörterbuch eingetragen wird. Der Standard kennt dabei keine Vorschriften für die Art und Weise, in der das letztendlich erreicht wird.

In einer natürlichen Sprache kann ein Wort je nach Kontext verschiedene Bedeutungen haben. Forth erlaubt dies ebenfalls. In Forth wird dies durch verschiedene Wortlisten erreicht. Gleichlautende Worte mit unterschiedlicher Funktion werden so untergebracht, daß jedes in einer anderen Liste steht. Um ein Wort in eine bestimmte Liste zu compilieren, stellt Forth ein Standardwort zur Verfügung (definitions). Und es gibt andere Worte mit denen die Durchsuchung einer Reihe von Listen während der Compilation oder der Ausführung bestimmt werden kann (search order).

#### **Stacks**

Forth gebraucht Stapel (stack). Die Worte tauschen Daten über den Datenstapel (parameter or data stack). Praktisch alle Standardworte nehmen Argumente vom Stapel und lassen Ergebnisse dort.

Der Standard kennt einen zweiten Stapel für Rücksprungadressen (return stack). Forth verwaltet dort seine Rücksprünge, solange ein Wort auf tieferer Ebene ausgeführt wird. Dieser Returstack darf vorübergehend als Datenablage benutzt werden, sofern der Ablauf des Systems dadurch nicht beeinträchtigt wird.

#### **Speicher und Adressierung**

Die Stapel können im Speicher oder in einigen Fällen im Prozessor selbst liegen. Die Wörterbücher dürfen sowohl im ROM als auch im RAM liegen. Anderweitig nicht gebrauchter Speicherplatz steht für eine Applikation zur Verfügung, dies unterliegt Begrenzungen abhängig von der Art der Implementation. Die Verwaltung des Speichers ist ebenfalls abhängig von der Implementation, aber bei der Ausführung solcher Funktionen helfen Standardworte in einer portablen Weise.

Dieser Standard bestimmt in keiner Weise die Implementation von ANS-Forth auf Maschinen bezüglich Adressierungsschemata oder Wortlängen.

#### **Der Forth Interpreter - Compilation und Ausführung**

Der Interpreter führt sowohl ANS-Forth Systemoperationen wie Standardprogramme aus. Er vergleicht das nächste Wort des Eingabestroms mit denen in der gültigen Suchfolge. Ist ein gleiches Wort gefunden und das System im Interpreter-Status, wird es ausgeführt. Wird der Text nicht gefunden, versucht der Interpreter die

Umwandlung in eine Zahl und legt diese auf den Datenstapel. Gelingt dies nicht, besteht ein undefinierter Zustand (ambiguous condition).

Befindet sich das System im Compile-Status und ist es ein Sofort-Wort (immediate word), wird es ebenfalls ausgeführt, ansonsten wird die Semantik des Wortes an die gerade compilierte Definition angehängt. Der Interpreter versucht, undefinierten Text in eine Zahl umzuwandeln und dann Code in die Definition einzufügen, welcher diese Zahl auf den Stack legt sobald sie ausgeführt wird. Gelingt dies nicht, besteht ein undefinierter Zustand.

#### **Forth Prompt**

Ein ergonomisches Design verlangt ein positives Zeichen wenn das System die Anfrage verstanden und darauf reagiert hat. Forth erreicht dies durch ein "prompt" am Ende des Prozesses. Es signalisiert beides, erfolgreiche Ausführung und Bereitschaft für weitere Eingaben. Es ist alte Forth Tradition, dies durch die Buchstaben OK darzustellen.

(Chuck Moore meinte, ein prompt bedeute soviel wie: "Mach voran, Dummkopf - ich warte!" Dagegen sei das OK des Forth mehr wie die vornehme Art eines Butlers: "Ausgezeichnet mein Herr, kann ich sonst noch etwas für Sie tun?")

#### **Definierende Worte (defining words)**

Ein nützliches Forthkonzept in vielen Fällen sind "Defining Words". Mit diesem leistungsstarken Mechanismus kann Worten eine spezielle erweiterte Fähigkeit gegeben werden, ihrerseits neue Worte mit eigenem Verhalten zu erzeugen.

Es gibt mehrere definierende Worte im Standard. So erzeugen sie etwa Objekte wie Konstanten oder Variablen. Andere erzeugen (create) neue auf das System zugeschnittene definierende Worte. In einem definie-

## BASIS 15

### Besprechung des ANS-Forth Entwurfs - Stand 4/91

renden Worte gibt es immer ein Teilstück mit Compiliereigenschaften und eines mit Laufzeiteigenschaften. Der compilierende Teil erzeugt einen neuen Wörterbucheintrag einschließlich der vom Benutzer definierten Datenstruktur. Der Laufzeitanteil bestimmt, was das neu erzeugte Wort später machen wird. Die Kraft dieser Konstruktion beruht auf der Tatsache, daß der Laufzeitanteil unter anderem auch den Compiler wieder aufrufen kann, so daß ein so definiertes Wort wieder ein definierendes Wort sein kann. Durch solche definierenden Worte höherer Ordnung können Anwendungen kurz und bündig (concise) und fortgeschritten (advanced) geschrieben werden.

#### **Immediate Words**

Ein weiteres wichtiges Konzept des Forth-Compilers neben den definierenden Worten sind die Sofort-Worte. Auch während der Compilation werden sie sofort ausgeführt, statt compiliert zu werden.

Beliebige neue Worte dürfen als "immediate" gekennzeichnet werden. Dadurch kann der Programmierer den Compiler erweitern. Noch während der Compilation anderer Worte kann so ein Wort jede gewünschte Aufgabe erledigen - Auswahl aus Tabellen, spezielle Textbehandlung (parsing action), Berechnungen, usw. Effizienter Code und eine eigene Anwendungsgrammatik zu erzeugen wird so leicht gemacht.

#### **Massenspeicher**

Die Speicherung und der Zugriff auf Informationen wird in diesem Standard auf zwei verschiedene Arten ermöglicht. Entweder kann das vorhandene Betriebssystem des Rechners über ein File-Interface mitbenutzt werden, oder es werden "virtuelle Blöcke" benutzt. Diese Blöcke sind eine Art von virtuellem Speicher für das Forth Programmiersystem und vorteilhaft bei der Übertragung

von Programmen zwischen Computern. Die Blöcke werden über Nummern adressiert und haben einheitliche feste Größen.

Dieser Standard legt fest, daß ein ANS-Forth solche Blöcke zur Verfügung stellen muß, wenn überhaupt irgend ein Massenspeicher unterstützt wird. Ein File-Interface zum Betriebssystem des Rechners hingegen muß nicht unbedingt bereit stehen.

#### **Andere Eigenschaften**

ANS-Forth bietet Worte für die Verwaltung von Einträgen (items) auf einem Stapel, arithmetische und logische Operationen, Speicher schreiben und zu lesen, Zeichen von der Tastatur annehmen und Zeichenketten ausgeben, Zeichenketten im Speicher verschieben und all die gebräuchlichen Dinge beim Programmieren.

#### **Benutzung von Forth Programmiersystemen**

Forth Programmiersysteme sind interaktiv. Die Benutzung ist einfach. Über die Tastatur werden ein oder mehrere Forthworte eingegeben und mit Drücken der Returtaste abgeschlossen, worauf der Forth Interpreter diese nacheinander ausführt. Wenn Worte Argumente benötigen, müssen diese zuvor auf den Datenstapel gelegt worden sein. Dazu tippt man entweder die erforderlichen Werte als Ziffern ein, oder führt ein Wort aus, daß diese Werte dort ablegt.

Exakt das gleiche Ergebnis wie von der Tastatureingabe bekommt man, wenn die Texteingabe zuerst auf dem Massenspeicher abgelegt und vom Interpreter dann von dort gelesen wird (input stream).

#### **Weitere Abschnitte des BASIS15 Begriffsbestimmungen**

Nach den "concepts" folgen gut zehn Seiten Begriffsbestimmungen.

Der Interessierte findet dort genaue Angaben darüber was z.B. mit "addition" gemeint ist (...overflow is ignored) oder was eine "ambiguous condition" ist oder "cell" oder "execution token" (...value that identifies a definition), "floored division" oder schließlich "word set" (...The core word set contains the essential body of words in an ANS Forth programming system. It is the only 'required' word set).

#### **Rahmenbedingungen**

Danach werden zum Gebrauch erforderliche Rahmenbedingungen geschildert. So wird hier festgelegt, daß bestimmte Parameter je nach Absicht der Implementation zugeschnitten werden können. Es werden gut vierzig Parameter in der Liste aufgeführt, die wesentliche Aspekte eines Forthsystems beschreiben. Ein Standardsystem muß all diese Parameter dokumentieren und näher erklären. Das reicht von "address units" über "minimum search order" bis "structure of a word list". Festgelegt werden dann noch die Bereiche für eine ganze Reihe von typischen Bedürfnissen eines Forthsystems -- testing requirements, minima, parameter range restrictions, word names and definitions, data type compatibility, numbers, arithmetic use of flags, addressable memory, data space, transient areas, return stack, compilation, control flow, input stream, error conditions, ambiguous conditions.

#### **Übereinstimmung und Bezeichnung**

So habe ich mir "Compliance and Labeling" übersetzt. Hier wird jedenfalls bestimmt, wann ein Forth sich ANS-Forth nennen darf. Ich zitiere: "Ein Forth-Programmiersystem ist mit diesem Standard konform und darf als 'ANS Forth System' bezeichnet werden, wenn es in der Lage ist, jedes ANS Forth Programm zu compilieren und auszuführen, und die Ab-

## BASIS 15

### Besprechung des ANS-Forth Entwurfs - Stand 4/91

hängigkeit des Programms von einer bestimmten Umgebung (environmental dependencies) mit der Beschränkung des Systems auf eine bestimmte Umgebung verträglich ist (environmental restrictions).

#### Was ist ANS Forth?

##### System

Ein ANS Forth System definiert das gesamte Core Word Set wie in diesem Dokument beschrieben. Darüber hinaus darf ein ANS Forth System Worte aus jedem Optional Word Set und Extension Word Set enthalten, wenn diese sich wie beschrieben verhalten.

Dieser Standard schreibt jedoch nicht vor, daß alle Worte in direkter ausführbarer Form vorliegen müssen. Der Herausgeber kann sich dafür entscheiden, Wortdefinitionen nur als Quellcode beizufügen, auch wenn diese zum Core Word Set gehören. Dies ist erlaubt, solange ein klar dokumentierter Ablauf existiert, mit dessen Hilfe die Definitionen dem Wörterbuch angefügt werden können.

Ein ANS Forth System darf unstandardisierte Erweiterungen enthalten solange diese kein unstandardisiertes Verhalten bei ANS Forth Programmen verursachen, wenn diese die Umgebungsbedingungen des Systems erfüllen. Ein ANS Forth System darf keine Worte benutzen die im Standard und in der üblichen Suchreihenfolge zu finden sind, aber sich anders als darin beschrieben verhalten.

Im Anschluß an diese Bestimmungen wird eine Liste von Aussagen angegeben, die ein ANS Forth System über sich machen sollte. Diese rund vierzig "items" beschreiben das jeweilige System in seinen Grundzügen und Eigenarten.

Und schließlich bemüht sich das TC im Dokument BASIS15 an dieser Stelle, möglichst viele der bereits existierenden Forthsysteme unter den Hut "ANS Forth" zu kriegen. Selbst

ein Forth, das nicht mal das Core Word Set komplett beinhaltet, darf sich immerhin noch "ANS Forth System Subset" nennen. Die gewachsene Vielfalt der Systeme auf unterschiedlichsten Maschinen und Betriebssystemen bleibt so in der gewohnten Weise erhalten. Aber man darf hoffen, daß wenigstens klar wird, welche Art Forth man vor sich hat - und was zu tun ist um Programme zu portieren - was schließlich das einzige Ziel des Standards ist.

Die Integration dieser Vielfalt gelingt mit den Zauberworten "Abhängigkeit der Programme von einer bestimmten Umgebung" (environmental dependencies) und "Beschränkung des Systems auf eine bestimmte Umgebung" (environmental restrictions).

Bestimmte Forthsysteme haben Beschränkungen in den bereitgestellten Worten oder sogar Einschränkungen der Minimalforderungen aufzuweisen. Trotzdem ist die Bezeichnung "ANS Forth System" grundsätzlich erlaubt, solange die verwendeten Worte sich wenigstens standardgemäß verhalten.

Zur Kennzeichnung des implementierten Umfangs wurden die "Word Sets" geschaffen. Jedes Word Set hat die zwei Teile Pflicht und Kür (word set and extension). Das nötigste ist im "core word set" zusammen gefaßt worden. Alles andere findet man in den "optional word sets". Diese wiederum bestehen jeweils aus der Grundmenge, welche komplett implementiert sein muß und den "extensions".

Um die Bezeichnung "ANS Forth System with the [option] word set" führen zu dürfen, muß das optionale Word Set vollständig implementiert sein. Für die Bezeichnung "ANS Forth System with the [option] Extension word set" müssen darüber hinaus auch die Erweiterungen vollständig vorhanden sein.

Ein System darf als "ANS Forth System Subset" bezeichnet werden, wenn es Teile des Core Word Set

enthält und diese dem Standard entsprechen. Aber es darf kein Wort aus dem Core ein unstandardisiertes Verhalten haben.

##### Programm

Ein Programm ist konform mit dem Dokument und darf sich "ANS Forth Programm" nennen, wenn es ausschließlich auf Worte des Core Word Sets aufbaut in einer Weise, wie es der Standard erlaubt. Wird ein ANS Forth Programm geladen, muß es angeben, ob hinterher noch ein ANS Forth System besteht. Und es sollte Stellung nehmen zu den folgenden Punkten: Minimalausstat-



tung des Operator-Terminals, nötige Ressourcen und gegebenenfalls Umgebungsabhängigkeiten.

Die Abhängigkeiten von der Umgebung müssen näher bezeichnet werden. Es sind dies Einschränkungen in den Erfordernissen für das Testen, Minima des Systems oder Parameterbeschränkungen.

Speziell beschrieben werden dabei die folgenden Punkte: Textausgabe wird über fixierte Speicherstellen ohne Möglichkeiten der Parametrisierung abgewickelt, arithmetischer Gebrauch von Flags, Shiftoperator

wird bei der Multiplikation benutzt ( $2 * D2*$ ), Shiftoperator wird bei der Division benutzt ( $2 / D2/$ ), Abhängigkeit von nicht dargestellten Zeichen (ACCEPT EXPECT), Verwendung von Kontrollzeichen oder Attribut-Bits (EMITTYPE), Verwendung von mehr als 7-Bit-Zeichen (EKEY), Verschachtelung von Eingabedateien über mehr als 8 Ebenen oder Zeilen länger als 128 Zeichen, Verwendung von Assembler-Code.

Wenn ein ANS Forth Programm zusätzliche Worte benötigt, muß dies angegeben werden. Alles zusätzlich benötigte muß einzeln aufgeführt werden. Beispiel: "Dieses ANS Forth Programm benötigt das Floating-Point Word Set sowie die Worte FSINCOS und FATAN2 aus dem Floating-Point Extension Word Set."

Interessanterweise kann ein Standardprogramm ein Standardwort umdefinieren zu einem Nicht-Standardwort - und immer noch ein Standardprogramm sein - denn es kann auf jedem Standardsystem laufen. Aber das Ergebnis (standard system plus standard program) ist ein Nicht-Standardsystem und muß als solches kenntlich gemacht werden!

### Zeichensatz

Natürlich regelt das Dokument auch, welcher Zeichensatz Verwendung findet. Im Anhang C ist er näher bezeichnet und abgebildet. Es ist natürlich der ASCII-Zeichensatz (ANS X3.4-1974 und ISO-646-1983). Selbstredend mit der notorischen Blindheit der Amerikaner für Zeichen aus anderen Sprachen: ohne Umlaute. Tröstlich ist, daß nur die Zeichen hex 20-7E festgelegt sind.

### Obsolet

Ausdrücklich wird hingewiesen auf Worte, die inzwischen obsolet geworden sind, obwohl sie früher viel gebraucht wurden. Sie wurden deshalb zwar noch in das core extension word set aufgenommen, sind aber für

die Löschung in zukünftigen Revisionen des Standards vorgesehen. Es sind dies die Worte: COMPILE CONVERT EXPECT SPAN [COMPILE]

POSTPONE übernimmt die Rolle von COMPILE und [COMPILE] NUMBER ersetzt CONVERT und ACCEPT wird statt EXPECT verwendet. SPAN wird ersatzlos gestrichen. Begründet wurde dies so: Die Unterschiede von COMPILE und [COMPILE] verstehe kein Mensch (richtig!); CONVERT sei schlecht faktorisiert; SPAN sei einfach nicht nötig gewesen bzw Ergebnis schlechten Stils; EXPECT habe ein unmögliches Benehmen.

Natürlich wird dies näher begründet. Doch das ist fast schon ein Beitrag für sich, diese Diskussion über Vor- und Nachteile aufleben zu lassen, und soll hier nicht mit abgehandelt werden.

### Liste der Word Sets

Core, Double Number, File Access, Block, Floating Point, Programming Tools, Error Handling, Search Order, String, Locals, Memory Allocation.

### Word Sets

Mit der Einführung von "word sets" will man dem starken Bedürfnis der verschiedenen Forth Programmierer entgegen kommen, ihr spezielles Forth für den jeweiligen Zweck zur Verfügung zu haben. So wird ein Wissenschaftler mehr Wert auf ein Floating Point word set legen als jemand der einfache Einbausteuerungen fabriziert (embedded controllers) und dort nur ein sehr mageres System gebrauchen kann.

Und aus diesen Anforderungen des Marktes entsprang die Erkenntnis, etwas wie einen Grundwortschatz zu formulieren und auf diesem dann weitere Schichten je nach Bedarf aufzubauen. Dabei stellte sich heraus, daß es sowohl Erweiterungen als auch echte Ergänzungen

gab, welche wiederum besondere Erweiterungen zuließen usw. So wurde das "Required Core Word Set" definiert und dazu einige "Core Extensions". Sodann die "Optional Word Sets" mit deren "Optional Word Set Extensions".

### Das Core Word set

Umfasst praktisch das, was bereits im Forth-83 bestimmt wurde plus einigen Neuerungen sowie Streichungen einzelner Worte. Manches wurde einfach nur in ein anderes Word Set ausgelagert. Dabei wurden aber keine prinzipiellen Dinge geändert. Der Übergang ist also verträglich zu bewerkstelligen. Auch das volksFORTH wird, so gesehen, weiterhin zu gebrauchen sein.

### Das Double Number Word Set

Darin finden sich Worte wie

$D+ D2* D.R M*/ M+$

und dererlei Operationen mit Werten aus Doppel-Zellen.

### Das File Access Word Set

Wenn es vorhanden ist, muß auch das Block Word Set vorhanden sein. In diesen beiden Sets wurde die Terminologie der Zugriffe auf Massenspeicher geregelt. Dabei wird über File-Identifikations-Werte auf dem Stack operiert (fileid, kurz: fid). Worten, die einen String brauchen, wie zB INCLUDE wird dieser mittels Adresse und Längenangabe auf dem Stack übergeben (counted string). Das ist konsequent UPN und nicht ständig auf neue Parser angewiesen. Man gebraucht es so: `... S" filename" INCLUDE ...` Dabei wird auch gleich klar, warum S" neu geschaffen wurde. Es legt Text aus dem Inputstream in einen eigenen Puffer und die Parameter dazu auf den Stack. Im Block Word Set finden sich die alten Bekannten BLOCK BUFFER FLUSH SAVE-BUFFERS UPDATE wieder. In der Extension dazu

## BASIS 15

### Besprechung des ANS-Forth Entwurfs - Stand 4/91

Worte wie zum Beispiel LIST LOAD SCR THRU.

#### **Das Floating Point Word Set**

Es ist recht umfangreich und enthält alle F-Operatoren für die Stackmanipulation, sowie Arithmetik und Trigonometrie. Da ich darin überhaupt nicht zu Hause bin, enthalte ich mich jeder weiteren Würdigung dieses Kapitels und hoffe, es findet sich ein Autor, der gerade dieses Thema einmal gründlich darstellt.

#### **Das Programming Tool Set**

Bietet Hilfen für den Programmierer. Hierin ist der Grundwortschatz leer - es gibt also eigentlich nur ein Extension Set. Da hat halt jeder so seine eigenen Ideen, was brauchbar sein könnte. In der Sammlung sind S. ? ;CODE BYE DUMP FORGET KEY? WORDS etc.

#### **Das Error Handling Word Set**

Dies ist eine Neuerung. Es enthält die zwei Worte CATCH und THROW und keine Extensions. Damit sei ein zuverlässiges Verfahren für die Fehlerbehandlung gegeben ohne sich mühsam mit Errorflags auf dem Datenstapel herumschleppen zu müssen; was in der Tat ein lästiges Verfahren war. CATCH und THROW arbeiten mit Hilfe des Returnstack im Sinne eines "non-local return". Das Verfahren ist inzwischen üblich auch in Sprachen wie C, die Namensgebung wurde aus LISP entlehnt. Die bekannten Worte ABORT und ABORT" können damit portabel werden.

#### **Das Search Order Word Set**

Es werden Primitivfunktionen für Wortlisten bereit gestellt. Die Art und Weise solche Listen zu erzeugen und zu Handhaben sind in den Forthsysteme FIG-Forth, Forth-79, poly-FORTH, und Forth-83 alle gegensei-

tig inkompatibel. Die komplette Liste aller Unverträglichkeit der einzelnen Hersteller ist noch länger. Daher der neue Ansatz, nur ein "primitiv tool" anzubieten, aus dem sich verschiedene Schemata je nach Bedarf bauen lassen. Damit soll eine Grundlage für die Portabilität geschaffen werden. Acht Worte werden angeboten, darunter DEFINITIONS FORTH ALSO ONLY und dann einige Worte wie SET-ORDER GET-ORDER etc. welche alle mit "word list identifier" (wid) per Stack arbeiten. So ein "wid" kann dann je nach System bedeuten ein Index in eine Wortliste, eine Adresse solch eines Datenfeldes, das "execution token" einer versiegelten Liste im Stile des Forth-83, die "link field address" der ersten Definition einer Liste oder irgend was anderes. Beim durchforsten dieser Verfahren und der Suche nach gemeinsamen Ansätzen erkannte man dann das eigentliche Grundproblem. Man mußte die Reihenfolge der Wortlisten retten/restaurieren können. Dies war nun mit den SET- bzw. GET-ORDER Primitiven gegeben. Beibehalten wurde die "compilation word list" CURRENT in die alle neuen Definitionen eingetragen werden. Fallen gelassen wurde die Liste CONTEXT, weil dahinter inzwischen nicht mehr nur eine weitere Liste sondern eine Suchreihe steckte und der ursprüngliche Sinn so in die Irre führt.

#### **Das String Word Set**

Es ist fünf bekannte Worte kurz: -TRAILING /STRING CMOVE CMOVE>COMPARE wobei /STRING ein natürlicher Faktor aus WORD ist, welcher so endlich einzeln zur Verfügung steht - es setzt die Parameter eines String um n Zeichen weiter. Neu und einziges Wort der Extension ist LEX und stellt einen textvergleichenden Parser dar.

#### **Das Locals Word Set**

Es ist noch kürzer: TO (LOCALS) sind die beiden Worte dort. Extensions: keine. (LOCALS) ist ein Primitive zur Implementation lokaler Variablen, jedoch nicht dazu geeignet, selbst direkt solche zu erzeugen. Eine ganze Reihe von Regeln muß dabei beachtet werden. Diese Regeln allein umfassen mit Kommentar zwei Seiten. Ziel ist es, damit das Wort LOCAL in portabler Weise erzeugen zu können.

#### **Das Memory Allocation Word Set**

Es ist ebenfalls kurz - vier Worte: ALLOCATE FREE RESIZE AVAILABLE. Damit werden zusammenhängende Speicherteile bereitgestellt oder an das System zurückgegeben, umgewandelt oder abgefragt.

#### **Neue Technologie**

Wie man an den Word Sets sehen konnte hat der Standard einige Worte und Praktiken neu aufgenommen, die immer häufiger in der allgemeinen Verwendung gefunden wurden. Außerdem wurden zur Erleichterung der Portabilität neue Worte speziell entwickelt. Das Glossar der Neuen im Grundwortschatz macht die wichtigen Entwicklungen deutlich und die Unterschiede zum Forth-83 klar. Dies soll an anderer Stelle gezeigt werden.

#### **Annex**

Das Dokument BASIS15 ist hier eigentlich zu Ende. Es folgt aber noch der Anhang A-J. Darin werden verschiedene Stellungnahmen, Kommentare und Tabellen geboten.

Einiges ist trivial aber für einen Antrag auf Zulassung als ANS wohl nötig (...singel cell data types are transferred from the stack to memory by ! and from the memory to the stack by @ ...). So zu lesen im Anhang A.

## BASIS 15

### Besprechung des ANS-Forth Entwurfs - Stand 4/91

Die Sammlung aller Fußnoten in Anhang B ist lehrreich zu lesen; hier finden sich auch hübsche Übersichten zu besonderen Problemen wieder. So die Kontrollstrukturen inklusive CASE. Auch die Kontroverse ACCEPT gegen EXPECT oder POSTPONE gegen COMPILE ist hier übersichtlich nachzulesen. Praktisch all das Besondere am ANS FORTH findet sich hier dargestellt - Eine Fundgrube.

Anhang C (Character Set) und D (Perspectives) geben nichts Besonderes her. Die "Perspectives" preisen halt die Vorzüge von Forth. Argumentationshilfen halt, falls man das

braucht - dann aber gleich seitenweise und gut gemacht finde ich.

Anhang E (Conformance Guide) war noch leer. Im Anhang F werden Ergänzungen, Ersatz und Streichungen gegenüber Forth-83 übersichtlich dargestellt und begründet. Der Vergleich mit Forth-79 fehlte noch (Anhang G). Weiter folgen noch Anleitungen zur Implementation (Anhang H) und zur portablen Programmierung (Anhang I) sowie ein Index aller Worte des Standards (Anhang J)

(Anmerkung: Das Original BASIS15 aus den USA liegt vor im Format für WINWORD - da mußte ich leider passen. Ich schreibe immer

noch mit WORD3.xx oder so. Ein Freund war so nett, das Dokument in ein einfaches ASCII-Format zu verwandeln. Dabei blieben so Sachen wie bunte Wörter und Kästen um Texte, fettgedrucktes und so weiter auf der Strecke, aber der pure Text wurde verfügbar. Interessierten schicke ich gern so eine Diskette zu.)

#### Literaturangabe:

Forth Programming System X3J14 Basis Dokument, revision: BASIS15 ANS X3/X3J14 Technical Committee 1991

Autor: Michael Kalus



# FFORTH für Atari ST

```
File Marken Editieren Block Parameter Speziell 17:51:22
-----
Sp 1 2 120
*****
Julianengen nach der BOUNDARY-SCANNING
*****
value att?
fvalue att.x
fvalue att.y
fvalue dx/2
fvalue current.x
fvalue current.y
%0.83 fvalue eps to eps

: ask.c cr ." Eingabe von C : " ask.complex
to cy to cx ;

: f> f<= not ;
: limit.attractor? ( -- )
att.x %100 f> att.y %100 f> or ;

needs PAUSE : pause ; immediate

find.attractor
```

**FFORTH- das Profi-Entwicklungssystem von Jörg Plewe.**

FFORTH ist eine sehr schnelle Implementation, die echten relokatabelen Maschinencode erzeugt. Ideal daher für Stand- Alone-Applikationen.

Compilieren aus dem Speicher, dadurch sehr kurze Turnaround- Zeiten.

Direkter Austausch des Quelltexts zwischen Compiler und Editor, dadurch ist ein extrem bequemes Programmieren möglich.

- Volle Unterstützung von AES, VDI, XBIOS und LINE A.
- Floatingpoint incl. 68881-Unterstützung.
- integrierter Assembler, Disassembler.
- Source (Forth)leveldebugger
- Online-Hilfen für über 750 FORTH-Wörter
- lokale Variablen und Multitasker
- maugesteuerte Oberfläche mit integriertem GEM-Editor (schneller als T....s!!)
- 250 seitiges Handbuch

**FFORTH-System: 248 DM**  
Demodisk: 10 DM

Bild: integrierter GEM-Editor Edwin 0.9

Außerdem im Angebot: Modulatoren, Umschaltbox U2, Virenkiller VIRENTOD, Grafikprogramm Star Designer, Datenfinder RETRIEVE, Echtzeitverschlüsselung TOP SECRET, Musikprogramm Soundman, Schachprogramme Deep Thought und DPE, Entwicklungspaket FForth und anderes mehr. Fordern Sie Infos an!

**Versandbedingungen:** Inland: Nachnahme 8,- DM Porto/VP; Vorkasse 4.50 DM Porto/VP Ausland: Nur Vorkasse + 10 DM Porto/VP



Stachowiak, Dörnenburg & Raeker GbR - Burggrafenstr. 88 - 4300 Essen 1  
Tel: 0201/27 32 90 oder 71 0 18 30 - FAX: 0201/71 0 19 50  
NL: Jolka Computing - Postbus 8183 - NL-6710 AD Ede

# Der dynamische Stringbuffer

Vorgestellt wird die Implementation eines Datenbuffers, wie er auch in PC/FORTH von LMI eingesetzt wird. Dieser Buffer kann aufgrund seiner einfachen Definition in jedem beliebigen Forth-System eingesetzt werden.

**Schlüsselworte:**  
**Stringoperationen, Ringbuffer**

## Der dynamische Stringbuffer

Das System PC/FORTH von LMI enthält ein überaus nützliches Feature, das es wert ist, hier näher besprochen zu werden.

Die Rede ist vom *dynamischen Stringbuffer*. Bei diesem handelt es sich um einen Datenbuffer mit einer sich kontinuierlich ändernden Speicheradresse (daher der Name 'dynamisch'). Er arbeitet prinzipiell wie ein Ringbuffer und dient zur Zwischenspeicherung und Übergabe von Speicherbereichen, wie sie z.B. auch Strings darstellen.

Wie der nebenstehende Screen zeigt, läßt sich ein solcher dynamischer Buffer mit absolut minimalem Aufwand in jedem Forth-System implementieren. Anfang und Ende des für den Buffer im Lexikon reservierten Bereichs sind in den Konstanten *Left* und *Right* gespeichert. Die hier angesetzte Länge von 1024 Bytes dürfte für eine gleichzeitige Speicherung von gut 20 Strings ausreichen (Strings sind selten länger als etwa 50 Bytes).

Das Wort *STRBUF* wurde als Konstante definiert und liefert eine Adresse im Stringbuffer, an der Daten abgelegt werden können. Als Startwert für *STRBUF* wurde das linke Ende des für den Buffer reservierten Bereichs gewählt.

Sollen im Stringbuffer weitere Daten gespeichert werden, ohne bereits dort abgelegte Daten zu überschreiben, so muß mit dem Befehl *+STRBUF* ein entsprechend langer Speicherplatz für diese neuen Daten

```
Screen #01
\ Ladeblock                                     HH 23:00 01.05.91
HERE 1024 ALLOT DUP CONSTANT Left 1024 + CONSTANT Right
Left CONSTANT STRBUF

: +STRBUF ( #bytes -- )
  STRBUF OVER - DUP Left U<
  IF DROP Right SWAP - ELSE NIP THEN
  [ ' STRBUF >BODY ] LITERAL ! ; EXCISE Left Right

: STRCAT ( adr1 len1 adr2 len2 -- adr3 len3)
  DUP 3 PICK + DUP +STRBUF >R
  2 PICK STRBUF + SWAP CMOVE
  STRBUF SWAP CMOVE STRBUF R> ;

: STR! ( adr len ^str -- ) 2DUP C! 1+ SWAP CMOVE ;
: STRPCK ( adr len -- ^str) DUP 1+ +STRBUF STRBUF STR! STRBUF ;
```

reserviert werden. Dies geschieht ganz einfach dadurch, daß der Wert der 'Konstanten' *STRBUF* um eine entsprechende Anzahl von Bytes weiter nach links versetzt wird. Sollte dies nicht möglich sein, weil sich dadurch eine Adresse außerhalb des erlaubten Bereichs ergeben würde, so wird vor diesem Versetzen der Wert von *STRBUF* auf das rechte Ende des Buffers gesetzt.

Aufgrund dieser Funktionsweise ist klar, daß im Stringbuffer abgelegte Daten grundsätzlich nicht dauerhaft vor weiterem Überschreiben sicher sind. Der Buffer soll lediglich dazu dienen, Ergebnisse oder Zwischenergebnisse z.B. von Stringoperationen in Empfang zu nehmen. Für eine dauerhafte Speicherung müssen diese Resultate anschließend an eine geeignete Stelle im Lexikon verschoben werden.

Abschließend sollen hier noch zwei Beispiele für den Einsatz des Stringbuffers gegeben werden:

Die Routine *STRCAT* hängt zwei durch Adresse und Länge spezifizier-

te Speicherbereiche von insgesamt maximal 1024 Bytes aneinander und legt das Resultat im Stringbuffer ab. Durch den Einsatz des Stringbuffers in dieser Routine ist es ohne weiteres möglich, *STRCAT* wiederholt aufzurufen. Die Befehlsfolge

```
" ABC" COUNT STRCAT
" DEF" COUNT STRCAT
" GHI" COUNT STRCAT TYPE
```

wird als Resultat ABCDEFGHI ausgegeben. Man mache sich klar, daß dies nur funktionieren kann, weil jedes Zwischenergebnis der Verkettungen im Stringbuffer an einer neuen Adresse abgelegt wird.

Recht nützlich ist auch die Routine *STRPCK*, welche einen durch Adresse und Länge spezifizierten Speicherbereich als String mit führendem Countbyte im Stringbuffer ablegt.

**Autor: Heinrich Hohl**

# FORTH ganz praktisch

*FORTH muß sich nach wie vor viele Vorwürfe gefallen lassen, von denen einige sicher nicht ganz zu unrecht erhoben werden. Einer dieser Vorwürfe beinhaltet, daß FORTH eine Sprache für 'elitäre' Spielereien sei und keinen praktischen Nutzen habe, es sei denn in der Meß- Steuer- und Regeltechnik. Ein anderer, sehr häufig angeführter Nachteil der diversen FORTH-Umgebungen ist in den Augen vieler potentiell Interessierter die Tatsache, daß man in TurnKey Applikationen immer 'das ganze FORTH' mitschleppen muß. Vielen Anwendern und Programmierern stößt es mittlerweile recht sauer auf, daß der Trend zu 'größer, schneller, schöner' immer mehr Überhand nimmt. Gerade von dieser Entwicklung nehmen aber auch die modernen FORTH-Versionen wie F-PC keinen Abstand. Eine Applikation die 10 Mal auf dem Bildschirm ausgeben soll: "Im Rheinland ist's am schönsten...", und die dabei als TurnKey Applikation satte 250 KByte groß ist, sprengt einfach das Maß des Erträglichen.*

Das beide Vorurteile nicht stimmen müssen, will ich mit zwei kleinen, praktischen Werkzeugen zeigen, die aus einem echten Bedarf heraus entwickelt wurden. Beide Werkzeuge sind in der DFÜ Szene entstanden und haben, wie es sich für gute FORTH-Programme gehört, dort inzwischen als 'Public Domain' ihre Verbreitung gefunden.

## **Das erste Beispiel:**

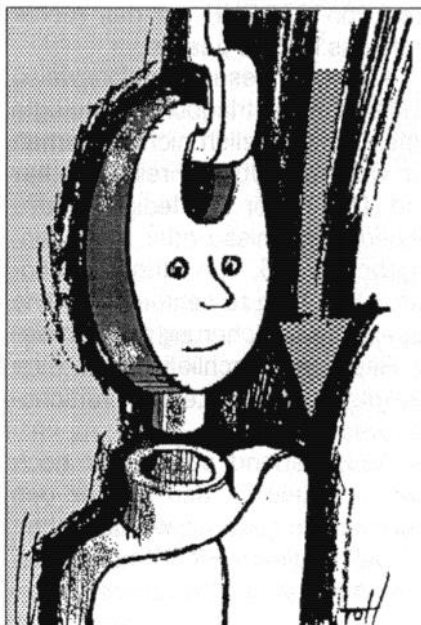
CHANGE.COM, ist aus folgender Aufgabenstellung heraus entstanden:

Die Kommunikation mit Mailboxen kostet natürlich Geld. Gerade dann, wenn man nicht auf eine Mailbox 'um die Ecke' zugreifen kann, reißen die diversen zusätzlichen Zeiteinheiten ein 23 Pfennig Loch um das andere in das Portomonaie. Es liegt also nahe, ab 18.00 Uhr die gewünschten Verbindungen herstellen zu lassen. Leider sind auf diesen naheliegenden Gedanken aber schon einige Tausend andere DFÜler gekommen. In vielen Boxen, vor allem in etwas größeren Städten, ist es oft sinnlos vor 22.00 Uhr überhaupt den Modem anzuschalten.

Der nächste Gedanke führt natürlich dahin, alle Transfers in Zeiten zu

legen, in denen die Telefonleitungen gewiß frei sind. Aber zu diesen Zeiten schlafen wohl die meisten Menschen auf unserem Erdteil einem neuen Arbeitstag entgegen. Ergo bleibt nur die Automatisierung.

Eine preiswerte Zeitschaltuhr schafft Abhilfe. Man kann seinen Rechner um 5.15 Uhr booten lassen und in der Autoexec anweisen, direkt in das Terminalprogramm zu springen. Das Terminalprogramm kann über 'Scripts' ohne das Beisein des Benutzers eine Mailbox anwählen und den gewünschten Datenaustausch vornehmen.



Einen 'Haken' hat die Sache. Wann immer man den Rechner also auch am Nachmittag, oder nach der Tagesschau, immer wird zuerst das Terminalprogramm starten. Das wird mit der Zeit zumindest lästig.

CHANGE.COM schafft hier Abhilfe. In die AUTOEXEC.BAT eingesetzt, startet CHANGE.COM Programme auf grund eines definierten DOS-Errorlevels, der abhängig von einer vorzugebenden Start- und Endzeit erzeugt wird. Als Beispiel mag meine eigene AUTOEXEC.BAT dienen. (s. S.17)

CHANGE.COM erzeugt den Errorlevel '1' wenn mein Rechner in der Zeit von 5.15 Uhr bis 6.00 Uhr eingeschaltet wird. In dieser Zeit wird das Terminalprogramm FrontDoor gestartet. Zu allen anderen Tageszeiten bootet mein Rechner 'ganz normal' im DOS.

Anwendungsmöglichkeiten zu CHANGE.COM lassen sich sehr viele denken. Und obwohl ich den oben erwähnten Spieltrieb durchaus weder leugnen kann noch will, ist CHANGE.COM ein nettes, kleines Werkzeug aus der Praxis, für die Praxis. Das hervorstechendste Merkmal von CHANGE ist aber seine Größe. Als



# FORTH ganz praktisch

```
AUTOEXEC.BAT
```

```
@ECHO OFF
PROMPT $P$G
LH c:\dos\keyb gr,,c:\dos\driver\keyboard.sys
LH c:\dos\fastopen c:=100
LH c:\editoren\qtsr\qtsr
PATH c:\;C:\dos;c:\dos\norton\;c:\dos\tools;c:\dos\bat;c:\dos\packer;
SET FD=c:\terminal\frodo
REM
REM -----
REM
CHANGE S=05:15 E=06.00
IF ERRORLEVEL 1 GOTO FRODO
IF ERRORLEVEL 0 GOTO NORM_BOOT
REM -----
REM FrontDoor, Fido-Net Terminal
:FRODO
CD C:\TERMINAL\FRODO
START_FD

:NORM_BOOT
CLS
```

TurnKey Applikation ist es gerade 2352 Byte groß, und das obwohl es bei seinem Aufruf einen halben Bildschirm voll Informationen über seine Syntax ausschüttet!

Selbstverständlich habe ich CHANGE.COM unter FORTH entwickelt. Entwicklungsumgebung war dabei das F-PC von Tom Zimmer. Kompiliert habe ich CHANGE.COM



allerdings mit dem TCOM Forth Target-Compiler. Auch TCOM ist von Tom Zimmer, und, wie alle Zimmerprodukte bisher, ausgesprochen empfehlenswert. TCOM erzeugt reinen Assemblercode und damit extrem kleine und schnelle \*.COM Files!

### **Das andere Beispiel,**

TAG\_IST.COM, ist ebenfalls aus dem 'DFÜ-Bedarf' heraus entstanden und ebenfalls unter F-PC entwickelt, aber mit TCOM kompiliert worden. Mit ganzen 2267 Bytes leistet TAG\_IST.COM eine dem CHANGE verwandte Arbeit und erzeugt Errorlevels in Abhängigkeit vom Wochentag.

Wozu sind Computer da, wenn nicht dazu um Routinearbeiten zu übernehmen? In der DFÜ sind solche Routinearbeiten das intervallmäßige Löschen von 'Databases', das Reorganisieren der Festplatte, die Zusammenstellung von Datenpake-

ten die nur wöchentlich verschickt werden und Vieles mehr. Auch im 'Hobbyalltag' lassen sich sicher einige 'Bedarfsfälle' für TAG\_IST.COM finden. Wie man mit diesem Werkzeug umgeht, soll eine kleine Datei mit dem Namen TEST.BAT verdeutlichen:

Soviel also zum praktischen Nutzen von FORTH-Kompilaten und zu der Behauptung, auch FORTH-Applikationen müßten zwangsläufig ins Uferlose wachsen.

Die beiliegenden Quelltexte CHANGE.SEQ und TAG\_IST.SEQ lassen sich, mit kleinen Änderungen, sofort mit F-PC kompilieren und testen. Unter ZF (ebenfalls von Tom Zimmer), oder unter F83 Systemen muß vor allem die Syntax der kleinen Assemblerrountinen überprüft werden.

Das  
**FORTH-Magazin**  
ist mittlerweile eine  
wissenschaftliche  
Zeitschrift - also  
für fachkundige  
Leser. Eine  
Gelegenheit für  
Veröffentlichungen  
und Darstellungen.  
Sie können zeigen  
was in ihnen  
steckt. Gefragt ist  
Grundsätzliches  
ebenso wie  
Ausblicke.  
Sachliche  
Information neben  
persönlicher  
Stellungnahme,  
klar gesagt und  
ohne Umschweife,  
Trends,  
Hintergründe,  
Werkzeuge,  
Techniken,  
Fragen, Antworten.  
Anzeigen.

Who is who  
und  
wer hat was

--

FORTH ganz praktisch

```
REM
REM      Beispiel Batch für 'TAG_IST.COM'
REM
REM -----

tag_ist

IF ERRORLEVEL 7   sonntag
IF ERRORLEVEL 6   samstag
IF ERRORLEVEL 5   freitag
IF ERRORLEVEL 4   donnerstag
IF ERRORLEVEL 3   mittwoch
IF ERRORLEVEL 2   dienstag
IF ERRORLEVEL 1   montag
IF ERRORLEVEL 0   ende

:montag
    nc
    GOTO ende

:dienstag
    excel
    GOTO ende

:mittwoch
    zf
    GOTO ende

:donnerstag
    fpc
    GOTO ende

:freitag
    telix
    GOTO ende

:samstag
    frodo
    GOTO ende

:sonntag
    auto
    GOTO ende

:ende
```

**FORTH-Magazin**

# FORTH ganz praktisch

```
\
\
\ CHANGE.COM
\
\ erzeugt den Errorlevel (DOS) '1' wenn die Systemzeit sich
\ innerhalb der anzugebenden Start - End Zeit befindet.
\
\ Syntax : CHANGE S=17:30 E=18:45
\
\         Dieser Aufruf würde den Errorlevel '1' erzeugen, wenn
\         das System in der Zeit von 17:30 - 18:45 bootet.
\
\ CHANGE.COM soll aus der AUTOEXEC.BAT heraus zu bestimmten
\ Tageszeiten bestimmte Programme direkt laden.
\
\
\
: ?alles_korrekt                                \ Überprüfung, ob die Eingabe
  TIB 1 + C@ ASCII S = IF                       \ in der korrekten Syntax
  TIB 2 + C@ ASCII = = IF                       \ CHANGE S=nn:nn E=nn:nn
  TIB 8 + C@ ASCII = IF                         \ erfolgt ist.
  TIB 9 + C@ ASCII E = IF                       \ Nicht 'benutzte' 'n' müssen
  TIB 10 + C@ ASCII = = IF                     \ mit '0' aufgefüllt werden.
  -1 THEN THEN THEN THEN THEN
;

: ?in_der_zeit                                  \ liegt die aktuelle System-
                                                \ zeit in der angegebenen Zeit ?

\ --- Extrahieren der Systemzeit ! -----
GETTIME                                         \ Systemzeit auslesen
DROP                                           \ Sekunden und Hundertstel sind unnötig
DUP                                           \ Stunden/Minuten duplizieren
255 AND                                       \ Minuten extrahieren
SWAP 255 /                                    \ Stunden extrahieren
                                                \ Die aktuelle Systemzeit liegt jetzt im
                                                \ Format MINUTE STUNDE auf dem Stack !
60 * +                                        \ und wird jetzt in eine Gesamt-Minutenzahl
                                                \ konvertiert.

\ --- Extrahieren der Startzeit ! -----
TIB 3 + C@ 48 -                               \ 1. Ziffernzeichen lesen, in Zahl umwandeln
  10 *                                         \ richtige Zehnerstelle schaffen !
TIB 4 + C@ 48 -                               \ 2. Ziffernzeichen
  +                                           \ Stunden zusammenaddieren
  60 *                                         \ in Minuten konvertieren
TIB 6 + C@ 48 -                               \ 3. Ziffernzeichen
  10 *
TIB 7 + C@ 48 -                               \ 4. Ziffernzeichen
  +                                           \ Minuten addieren
  +                                           \ und auf konvertierte Stunden aufaddieren
```

## FORTH ganz praktisch

```
DUP 0<    IF DROP 0 THEN    \ Minuten < 0 und größer 1440 (24:00)
DUP 1440 > IF DROP 0 THEN    \ kann es nicht geben !
```

```
\ --- Extrahieren der Endzeit ! -----
```

```
TIB 11 + C@ 48 -    \ 1. Ziffernzeichen lesen, in Zahl umwandeln
    10 *            \ richtige Zehnerstelle schaffen !
TIB 12 + C@ 48 -    \ 2. Ziffernzeichen
    +              \ Stunden zusammenaddieren
    60 *           \ in Minuten konvertieren
TIB 14 + C@ 48 -    \ 3. Ziffernzeichen
    10 *           \
TIB 15 + C@ 48 -    \ 4. Ziffernzeichen
    +              \ Minuten addieren
    +              \ und auf konvertierte Stunden aufaddieren
```

```
DUP 0<    IF DROP 0 THEN    \ Minuten < 0 und größer 1440 (24:00)
DUP 1440 > IF DROP 0 THEN    \ kann es nicht geben !
```

```
\ -----
\
\      konvertiert in Angaben aus Minuten liegen nun die Endzeit auf dem
\      TOS, darunter die Startzeit, und darunter die tatsaechliche
\      Systemzeit !!!
\ -----
```

```
ROT  DUP  ROT  <=    \ Zeiten miteinander vergleichen...
ROT  ROT  SWAP >=
AND                                     \ auf dem Stack bleibt nur ein Flag
                                         \ zurück !
```

```
;
```

```
\ -----
\
\      Ein kleines Bischen Assembler muß sein.
\ -----
```

HEX

```
CODE errorlevel1
MOV AH, # 4C    \ Funktion 4Chex im AH Register
MOV AL, # 01    \ und Errorlevel '1' in AL...
21 INT         \ ...und der DOS-Interrupt 21hex beendet das
END-CODE       \ Programm
```

```
CODE errorlevel0
MOV AH, # 4C
MOV AL, # 00
```

# FORTH ganz praktisch

```
21 INT
END-CODE

DECIMAL

: .hilfe
  CR 36 %SPACES ." CHANGE.COM"
  CR 35 %SPACES ." ====="
  CR 26 %SPACES ." Aufruf: CHANGE S=nn.nn E=nn.nn"
  CR CR
  20 %SPACES ." Der DOS-Errorlevel '1' wird zwischen der"
  CR 20 %SPACES ." Startzeit 'S' und der Endzeit 'E' erzeugt."
  CR
  CR 21 %SPACES ." Bitte beachten Sie, daß Sie 'S' und 'E'"
  CR 21 %SPACES ." in Großbuchstaben schreiben müssen !"
  CR
  CR 21 %SPACES ." Bitte beachten Sie, daß nach CHANGE nur"
  CR 21 %SPACES ." ein Leerzeichen stehen darf ! -----"
  CR 20 %SPACES ." ====="
  CR 21 %SPACES ." (co) Friederich Prinz, NRW"
  CR 21 %SPACES ." TCOM - FORTH Target Compiler"
;

\ =====

: change
  DECIMAL
  DOSIO_INIT
  DOS_TO_TIB
  .hilfe \ Ausgabe einer Syntaxbeschreibung
  ?alles_korrekt
  IF ?in_der_zeit
  THEN
  IF errorlevel1 \ Definitionen der DOS-Rückgabecodes !!!
  ELSE errorlevel0
  THEN
  BYE
;

\ //////////////////////////////////////
\
\ F-PC / TCOM Source,
\ Friederich Prinz
\ 80386-33 / 64 Cache
\ MSDOS 5.0
\ QEDIT, externer TSR-Editor
\ 08.04.1991
\
```



## FORTH ganz praktisch

```

                                4 +LOOP
                                THEN
                                DUP -1 IF DROP 0 THEN
;

: ?schalttag ( -- )              \ wenn das aktuelle Jahr ein
    GETDATE SWAP DROP           \ Schaltjahr ist, muß ab dem
    255 /                       \ 2. Monat ein zusätzlicher
    2 >                          \ Tag gezählt werden.
    ?schaltjahr
    AND
    IF 1 anzahl_tage +! THEN
;

: .meldung ( b -- )
    ." Tag ist " DUP 1 = IF ." Montag "      THEN
                    DUP 2 = IF ." Dienstag " THEN
                    DUP 3 = IF ." Mittwoch "  THEN
                    DUP 4 = IF ." Donnerstag " THEN
                    DUP 5 = IF ." Freitag "   THEN
                    DUP 6 = IF ." Samstag "   THEN
                    DUP 7 = IF ." Sonntag "   THEN
    ." --- DOS-Errorlevel ist : " .
;

\ =====

\ Auch hier braucht's etwas Assembler....

HEX

CODE errorlevel ( b -- )
    MOV AL, BL          \ BL hält das Byte auf dem Stack, das muß
    MOV AH, # 4C       \ nach AL (Errorlevel), AH bekommt die
    21 INT              \ Funktionsnummer 4Chex, und der Interrupt
END-CODE               \ 21 Hex beendet das Programm

DECIMAL

\ =====

: kalender
    decimal             \ hier wird dezimal gearbeitet
    dosio_init          \ Bildschirmausgaben initialisieren
    vorjahre>tage       \
    addiere_schaltjahre \ Auszählen der Tage seit dem
    monate>tage         \
                        \ 01.01.1991
    addiere_akt_tage    \
    ?schalttag          \
    anzahl_tage @ 7 MOD 1+ \ Berechnung des Kalenders : Anzahl
                        \ der angefallenen Tage durch 7,
                        \ aber nur der REST interessiert !
                        \ 'ohne Rest' = eine neue Woche =
                        \ Montag. 1+ bewirkt, daß der Monatg
```

```

\ mit '1' gezählt wird, Dienstag mit '2'
\ usw. fortlaufend...
DUP \ Der TOS wird zweimal gebraucht...
.meldung \ ...für die Meldung...
errorlevel \ ...und für den Errorlevel
bye \ ... das war's dann wohl....
;

\
\
\ //////////////////////////////////////
\
\ F-PC / TCOM Source,
\ Friederich Prinz
\      80386-33 / 64 Cache
\      MSDOS 5.0
\      QEDIT, externer TSR-Editor
\      08.04.1991
\
\ Mit den beiden Programmen wünsche ich viel Spaß

Autor: Friederich Prinz
```

## Problembeschreibung : CURSOR

FORTH-Kategorie:  
FORTH-Splitter

FORTH-Standard:  
FORTH 83

FORTH-System:  
volksFORTH  
systemspezifisch

**Schlüsselwörter:**  
**Cursor, Hardware**

Im volksFORTH rev.3.81.41 treten, bedingt durch die Vielfalt an Grafik-Karten für den PC, unter Umständen Probleme mit dem CURSOR auf.

Denn in dem meisten PCs verriet(en?) eine Hercules-Karte ihren Dienst, auf die einige Worte im EDITOR und im Kommandozeilen-Editor CED zugeschnitten sind. Die Worte CURON und CUROFF bestimm-

men das Erscheinungsbild des Cursors durch Zahlenwerte für CURSHAPE, die grafikartenABHÄNGIG sind. Hier bietet sich eine Änderung mit Hilfe des Video-Interrupts INT\$10 an, dessen Funktion \$3 im C-Register die aktuelle Start- und End-Zeile des Cursors wiedergibt.

Als Warnung: Im Zuge meiner Versuche zur Interruptprogrammierung hat (nach ungesichertem R-Register) ein V - aktuellen Screen editieren - dafür gesorgt, daß meine gesamte freie Festplattenkapazität der Datei EDITOR.SCR zugeschlagen wurde.

Mit dem Kommandozeilen-Editor CED hängt auch das zweite Problem zusammen: Wie man in SHOWCUR sieht, wird der Einfüge- oder Überschreibmodus der Kommandozeile durch das Erscheinungsbild des Cursors verdeutlicht. Da liegt es auf der

Hand, daß in (EXPECT die Wortfolge KEY DECODE SHOWCUR nach jeder Zeicheneingabe das Aussehen des Cursor in Abhängigkeit vom jeweiligen Modus festlegt und ein etwaiges, vom Anwender gesetztes ( top bot) CURSHAPE unwirksam machen !!

Soll dem nicht so sein, bleibt es dem Anwender unbenommen, den CURSHAPE auszulesen und in eine 2Variable mit 2! einzuspeichern und nach Veränderungen mit 2@ wieder auszulesen. Denn CURON/CUROFF werden oft verwendet und bringen bei absoluter Zuordnung nur Probleme und Verwirrung mit sich.

Hoffentlich können diese Hinweise zu einer endgültigen sauberen Problemlösung beitragen.

**Autor: Jörg Staben**



# Neues im Core Word Set

## Was ANS Forth im CORE WORD SET anders macht...

*Die hier aufgeführten Forthworte waren im Forth-83, dem bisherigen Standard, nicht bekannt - insofern sind sie im Standard "neu". Es handelt sich dabei aber sehr wohl um Worte, die aus der Praxis stammen und sich mehr und mehr als wichtige Instrumente durchgesetzt haben, so daß sie jetzt in den Standard aufgenommen worden sind.*

Zur Syntax des Glossars: Der NAME der Definition steht am Anfang der Abschnitte, gefolgt von der Aussprache, welche in Anführungszeichen gefasst ist. Dahinter gegebenenfalls Kennbuchstaben. D = defining word.

C = compiler word. In der nächsten Zeile steht das Stackbild ( vorher -- nachher ) in der für Forth üblichen Weise. Dann folgt die Definition des Befehls. Anschließend weitere Erklärungen und Beispiele zum Gebrauch.

Das Original BASIS15 lag als Ascii-Quelle vor. Sollten sie auf Fehler stoßen, sind diese vermutlich auch in der Quelle vorhanden. Ich bitte um Mitteilung zur Weitergabe an die Autoren des BASIS15. Die Bearbeitung beschränkte sich auf das Erscheinungsbild. Ich hoffe sehr, daß dabei nichts "verschluckt" wurde. mk

: "colon" D  
( "name" -- sys1 )

Parse name delimited by a space, ignoring leading delimiters. Create a dictionary entry for name with the execution semantics defined below. Enter compilation state. The newly created word definition for name cannot be found in the dictionary until the definition is completed. name is called a "colon definition".

sys1 is balanced by the corresponding ; or ;CODE.

name Execution: ( -- ) ( R: -- sys2 )

Save implementation-dependent information about the definition that called name. Typically, the execution semantics of name are expanded by compiling additional words into the definition.

Typical use: : name ... ;

See also: 3.6 The Forth Interpreter-Compilation and Execution, 3.7 Defining Words, 5.6 Compilation

Definition of colon was changed to NOT alter the search order. Gist of the argument: In most cases, the change has no effect -- MYVOC DEFINITIONS sets MYVOC as both the current and context word lists. In systems that allow extensive search orders, the behavior of colon is very undesirable.

> NUMBER "to-number"  
( ud1 c-addr1 u1 -- ud2 c-addr2 u2 )

ud2 is the result of converting the characters within the character string specified by c-addr1 u into digits, using the number in BASE, and accumulating each into ud1 after multiplying ud1 by the number in BASE. Conversion continues until a character that is not convertible is encountered or the string is entirely converted. c-addr2 is the location of the first unconverted character or the first character past the end of the string if the string was entirely converted. u2 is the number of unconverted characters in the string. An ambiguous condition exists if ud2 overflows.

See also: 4.0550 number conversion

ACCEPT  
( c-addr +n1 -- +n2 )

Receive a string of at most +n1 characters. An ambiguous condition exists if +n1 is zero or greater than 32,767. Display graphic characters as they are received. A Standard Program that depends on the presence or absence of non-graphic characters in the string has an environmental dependency. The editing functions, if any, that the system performs in order to construct the string are implementation-defined. Input terminates when "return" is received. When "return" is received, nothing appended to the string, and the display is maintained in an implementation-defined way. +n2 is the length of the string stored at c-addr.

Previous standards specified that collection of the input string terminates when either a "return" is received or when +n1 characters have been received. Terminating when +n1 characters have been received is difficult, expensive, or impossible to implement in some system environments. Consequently, a number of existing implementations do not comply with this requirement. Since line-editing and collection functions are often implemented by system components beyond the control of the Forth implementation, this standard imposes no such requirement. A Standard Program may only assume that it can receive an input string with ACCEPT or EXPECT. The detailed sequence of user actions necessary to prepare and transmit that line are beyond the scope of this standard. See H.2 ACCEPT and EXPECT MB

## Neues im Core Word Set Was ANS Forth im CORE WORD SET anders macht...

Specification of positive non-zero integer counts (+n1) for ACCEPT allows some vendors to continue their practice of using a zero or negative value as a flag to trigger special behavior. Insofar as such behavior is outside the standard, standard programs cannot depend upon it, but the TC doesn't wish to preclude it unnecessarily. Since actual values are almost always small integers, the TC doesn't feel any functionality is impaired by this restriction, and it is the policy of the TC to avoid impacting existing usage where possible. ER LZ

### ALIGN

( - )

If the address of the next available data space location is not aligned, reserve enough padding space to align it.

See also: 5.4.1 Data space

ALIGN and ALIGNED may be necessary to build and traverse data structures built with C, in a cell-aligned implementation. GS/265

### ALIGNED

( addr -- a-addr )

a-addr is the first aligned address greater than or equal to addr.

ALIGN and ALIGNED may be necessary to build and traverse data structures built with C, in a cell-aligned implementation. GS/265

Add the size of a cell, specified in address units, to addr1, giving addr2.

### CELLS

CELL+ "cell-plus" ( addr1 -- addr2 )

( n1 -- n2 )

n2 is the size, in address units, of n1 cells.

### CHAR "care"

( "ccc " -- char )

Parse characters ccc delimited by a space, ignoring leading delimiters. Put the integer value of its first character onto the stack.

Typical use: ... CHAR A CONSTANT "A" ...

CHAR+ "care-plus"

( addr1 -- addr2 )

Add the size of one character, specified in address units, to addr1, giving addr2.

### CHARS "cares"

( n1 -- n2 )

n2 is the size, in address units, of n1 characters.

### ENVIRONMENT? "environment-query"

( c-addr u -- value true ) (known) or

( c-addr u -- false ) (unknown)

c-addr is the address of a character string and u is the string's character count. u may have a value in the range from 0 to an implementation-defined maximum which may not be less than 31. The character string should contain a keyword from the table below to be checked for correspondence with an attribute of the present environment. If the system treats the attribute as unknown, the returned flag is false otherwise, the flag is true and the value returned is of the type specified in the table for the attribute queried.

String Value type Interpretation:

/ALIGN	n	alignment granularity, in address units, of an aligned address.
/CHAR	n	size of a character in address units
/COUNTED-STRING	n	maximum number of characters in a counted string
/DATA-SPACE	n	size of the aligned data space in address units
/HOLD	n	maximum size of a pictured numeric output string in address units
/PAD	n	size of the scratch area pointed to by PAD, in address units
/TIB	n	size of the text input buffer in address units
ADDRESS-UNIT-BITS	n	Size of one address unit in bits
BLOCK	flag	block word set present
BLOCK-EXT	flag	block extension word set present
CORE	flag	core word set present
CORE-EXT	flag	core extension word set present
DOUBLE	flag	double number word set present
DOUBLE-EXT	flag	double extension word set present
FILE	flag	file word set present

*the almighty FORTH:*



*für alle Atari ST/TT*

## Kleinanzeigen:

### Angebote:

Super8™-FORTH in Silicon

- FORTH im 8Kbyte On-chip ROM
- Transparente Entwicklungsumgebung kompiliert ROM-fähigen Code
- Zugriff auf alle On-chip Ressourcen
- ausführliches Handbuch
- Quellcode von S8-Assembler und S8-FORTH

Muster 0887520PSC bei:  
FORTHWORKS Ulrike Schnitter,  
Tel.:089-3103385

Es gibt noch einige Tagungsbände der FORTH '91, Preis: 20 DM inkl. Versand. Bestellungen bitte an das FORTH-Büro.

F-PC Zimmer FORTH Dokumentation jetzt erhältlich. Erweiterte Informationen. Kosten der drei Broschüren DM 80,- (für FORTH-Mitglieder DM 70,-). Vorkasse. Arndt Klingelberg, Straßburger Straße 12, W-5110 Alsdorf, Tel. 02404/61648.

### Suche:

Ronald Zech: FORTH 83  
T.J. Zimmer: TINY-PASCAL in fig-FORTH, Mountain View Press, Januar 1981, auch leihweise  
Volker Pohlens, Lomonossowallee 41/48, O-2200 Greifswald

Zwei gebrauchte 8 Zoll Diskettenlaufwerke, lauffähig.  
Peter Dinies, Metzstraße 38,  
W-2300 Kiel 1, Tel. 0431/1 32 39

Die Entwicklung von bigFORTH bleibt nicht stehen. Wenn Sie diese Zeilen lesen, wird bigFORTH Version 1.10 bereits erhältlich sein. Neben den bisherigen Features bietet die Version 1.10 folgende Neuheiten:

- Separate Compilation, kleine Applikationen durch einen Linker; 68030/68882-Assembler/Disassembler; History-Buffer für die Kommandozeile; eingebundener Streamfile-Editor, volle Unterstützung des TTs...

Natürlich bietet bigFORTH auch sonst alles, was ein gutes FORTH so braucht:

- **Mächtiger Compiler:**  
32-Bit-System, Native-Code, voll relokatable, weitgehend FORTH-83-kompatibel, nutzt Screen- und Streamfiles; relokatable Turnkey-Applikationen, denen man FORTH nicht mehr ansieht, sind problemlos machbar...
- **Vielseitige Tools:**  
68030-Assembler/Disassembler, Multitasker, sourcefähiger Decompiler, Source-Level-Debugger (Single-Step und Trace, beliebig viele Breakpoints), Post-Mortem Dump und Returnstack-Trace bei allen Fehlern, resetfest (für den Ausstieg aus der Endlosschleife), Multiwindow-Editor für Screen- und Streamfiles (natürlich unter GEM), beliebiger anderer Editor nutzbar...
- **Umfangreiche Libraries:**  
Komfortables Fileinterface, Druckertreiber, vollständige Betriebssystemanbindung (GEMDOS, BIOS, XBIOS, GEMAES, GEM-VDI und Line-A-Grafik), Floating-Point-Arithmetik, leistungsstarkes Memory Management, High-Level-GEM-Library, Turtle Graphic und mehr...
- **Transparenz:**  
Sämtliche Sourcen (einschließlich Kernal) auf Diskette, das modulare System ersetzt den Target-Compiler...

Die ausführliche deutsche Dokumentation (über 250 Seiten Handbuch) wird Anfängern ihre ersten Schluckbeschwerden bei dem großen Bissen beseitigen, aber auch für alte Hasen von Vorteil sein...

Und das alles nur für<sup>1</sup> DM 200.-

Update von Version 1.00: DM 20.-

Für Vorsichtige gibt es eine frei kopierbare Demoversion (ohne Sourcen&Handbuch) für eine Schutzgebühr von DM 10.-

Bestellungen an: **Bernd Paysan,**  
Stockmannstr. 14  
D-8000 München 71  
Tel. 089/798557

<sup>1</sup>(alle Preise incl. Mehrwertsteuer und Versand)

## Neues im Core Word Set Was ANS Forth im CORE WORD SET anders macht...

FILE-EXT flag  
file extension word set present

FLOATING flag  
floating-point word set present

FLOATING-EXT flag  
floating-point extension word set present

FLOATING-STACK n  
If n = 0 floating-point numbers are kept on the data stack, otherwise n is the maximum depth of the separate floating-point stack.

FULL flag  
full compliance as defined in Section 6.1

LOCALS flag  
locals word set present

LOCALS-EXT flag  
locals extension word set present

MAX-D d  
the largest usable signed double number

MAX-FLOAT r  
the largest usable floating-point number

MAX-N n  
the largest usable signed integer

MAX-U u  
the largest usable unsigned integer

MAX-UD ud  
the largest usable unsigned double number

MEMORY-ALLOC flag  
memory allocation word set present

MEMORY-ALLOC-EXT flag  
memory allocation extension word set present

RETURN-STACK-CELLS n  
the maximum size of the return stack in cells

SEARCH-ORDER flag  
search order word set present

SEARCH-ORDER-EXT flag  
search order extension word set present

STACK-CELLS n  
the maximum size of the data stack in cells

STRING flag  
string word set present

STRING-EXT flag  
string extension word set present

WORDLISTS n  
the maximum number of word lists usable in the search order

### EVALUATE

(i\*w c-addr u – j\*w)

Save the current input stream specification. Make the string described by c-addr and u the current input stream and interpret its contents. When the input stream is exhausted, restore the prior input stream specification.

The TC is aware that this function is commonly spelled EVAL. However, there may exist variations in its imple-

mentation that would suffer by defining the word as is done here. Additionally, EVALUATE is more readable and explicit. There was some sentiment for calling this INTERPRET, but that too would have undesirable effect on existing code. The longer spelling was not deemed significant since this not a word that should be used frequently in source code. GB/207

Standard words which access the input stream must know where the input stream is. Currently, the input stream comes from TIB or from a block. By specifying that the EVALUATE input stream comes from TIB, we no longer have to worry about inventing yet a third source of the input stream. MT

### FM/MOD "f-m-slash-mod"

(d1 n1 -- n2 n3)

Divide d1 by n1, giving the floored quotient n3 and the remainder n2. Input and output stack arguments are signed. An ambiguous condition exists if n1 is zero or if the quotient lies outside the range of a single-cell signed integer.

```
: /_MOD ( n1 n2 -- n3 n4 ) > R S>D R> FM/MOD ;
: /_ ( n1 n2 -- n3 ) /_MOD SWAP DROP ;
: _MOD ( n1 n2 -- n3 ) /_MOD DROP ;
: */_MOD ( n1 n2 n3 -- n4 n5 ) >R M* R> FM/MOD ;
: */_ ( n1 n2 n3 -- n4 ) */_MOD SWAP DROP ;
```

See also: 4.0420 floored division

This allows application programs to define the following floored division operators:

### INVERT

(mask1 -- mask2)

Invert all bits of mask1, giving mask2.

The word NOT was originally provided in Forth as a flag operator to make control structures readable. Under its intended usage the following two definitions would produce identical results:

```
: ONE ( flag -- ) IF ." true" ELSE ." false" THEN ;
: TWO ( flag -- ) NOT IF ." false" ELSE ." true" THEN ;
```

This was common usage from the dawn of Forth programming until the FORTH-83 standard was promulgated. That standard redefined NOT as a cell wide one's complement operation, functionally equivalent to the phrase -1 XOR. At the same time, the data type manipulated by this word was changed from a flag to a cell wide collection of bits and the standard value for true was changed from "1" (rightmost bit only set) to "-1" (all bits

## Neues im Core Word Set Was ANS Forth im CORE WORD SET anders macht...

set). Among other things, this destroyed the utility of NOT in its intended role, since under FORTH-83 the above two definitions are no longer equivalent. Many Forth users have declined to adopt the FORTH-83 version of NOT because they failed to see the advantage in changing all of their software only to compromise its readability and create a "programmer trap". The TC has concluded that usage of NOT cannot be standardized at this time. It's two meanings—flag inversion and one's complementation—are made explicitly available by the names 0= and INVERT, respectively. GB ed LZ

### M\*

( n1 n2 -- d )

d is the signed product of n1 times n2.

M\* is a useful early step in a calculation, going to extra precision at the moment you'll need it most. ER

### POSTPONE C

Compilation: ( "name" -- )

Parse name delimited by a space, ignoring leading delimiters. Append the compilation semantics of name to the current definition. An ambiguous condition exists if name is not found.

Execution: ( -- )

Perform the compilation semantics of name.

Typical use:

```
: ENDIF POSTPONE THEN ; IMMEDIATE
: X ... IF ... ENDIF ... ;
```

Remember, if name is not immediate, its compilation semantics are to append its execution semantics to the current definition. LZ

### S" "s-quote" C

Compilation: ( "ccc<"> -- )

Parse characters ccc delimited by " (double-quote). Append the execution semantics given below to the current definition.

Execution: ( -- c-addr u )

Return c-addr and u describing a string consisting of the characters ccc.

Typical use:

```
: X ... S" ccc" ... ;
```

### SHIFT

( w1 n -- w2 )

Perform a logical shift of n bit-places on w1, giving w2. If n is positive, shift the bits n places toward the high order bit. If n is negative, shift w1 toward the low order bits. Put zero into the places "uncovered" by the shift. Leave the result as w2.

### SM/MOD "s-m-slash-mod"

( d1 n1 -- n2 n3 )

Divide d1 by n1, giving the symmetric quotient n3 and the remainder n2. Input and output stack arguments are signed. An ambiguous condition exists if n1 is zero or if the quotient lies outside the range of a single-cell signed integer.

This allows application programs to define the following symmetric division operators:

```
: /-MOD ( n1 n2 -- n3 n4 ) >R S>D R> SM/MOD ;
: /- ( n1 n2 -- n3 ) /-MOD SWAP DROP ;
: -MOD ( n1 n2 -- n3 ) /-MOD DROP ;
: */-MOD ( n1 n2 n3 -- n4 n5 ) >R M* R> SM/MOD ;
: */- ( n1 n2 n3 -- n4 ) */-MOD SWAP DROP ;
```

See also: 4.0755 symmetric division

### UNLOOP C

( -- ) ( R: sys -- )

Discard the loop control parameters. The loop control parameters must have been available.

Typical use: : X ... DO ... IF ... UNLOOP EXIT THEN ... LOOP... ;

See also: 5.5 Return Stack

UNLOOP as a function has been called UNDO. UNLOOP is more indicative of the action nothing gets undone -- we simply stop doing it.

### [ "left-bracket" C

( -- )

Enter interpretation state. [ is an immediate word. Use of this word in interpret state is an error.

Typical use : X ... [ 4321 ] LITERAL ... ;

See also: 3.6 The Forth Interpreter -- Compilation and Execution, 8.1.2540 ]

### [CHAR] "bracket-care" C

Compilation: ( "ccc " -- )

## Neues im Core Word Set Was ANS Forth im CORE WORD SET anders macht...

Parse characters ccc delimited by a space, ignoring leading delimiters. Compile char, the integer value of the first character of ccc, as a literal.

Execution: ( - char )

Place char on the stack.  
Typical use: : X ... [CHAR] ccc ... ;

**Autor: Michael Kalus**

---

## Jahrestagung in München (Fortsetzung von Seite 7)

Nach dem Abendessen hörten wir noch vom Stand der Entwicklung unserer Mailbox, welche jetzt wieder in München beim Sysop Jens Wilke angesiedelt ist. Die einzelnen Beiträge sind übrigens demnächst im Tagungsband nachzulesen.

Der Samstag verlief wie geplant mit interessanten Beiträgen über Forthanwendungen. Sehr gefallen hat mir der Beitrag von Herwig Höfner, MPI für extraterrestrische Physik - wobei ich kaum sagen kann was mir besser gefallen hat: Die schönen Bilder der Nordlichter oder die schlaue Verwendung von Forth im Satelliten zur Versuchssteuerung bei der Feldmessung über den Polen der Erde zur Erforschung dieser Nordlichter. Ich will nicht alle Beiträge hier zitieren, das würde den Rahmen sprengen, der mir gegeben ist - interessant waren sie alle. Es war eine gute Wahl. Mich persönlich hat dabei besonders gefreut, wie selbstverständlich Forth inzwischen eingesetzt wird, ohne darüber noch besondere Worte zu verlieren. Es berichteten Klaus Flesch, Forth-Systeme (Swiss-FORTH-RTX), Dr. Egmont Woitzel, FORTECH (comFORTH für Microcontroller), Dr. Olaf Haeggqwist (RTX-2000-Entwicklungssystem), Birgit Steffenhagen, FORTECH (KI-

Methoden für die Lösung automatisierungstechnischer Aufgaben), Konstantin Vassiliev (Neural Networks Simulator Toolkit), Chris Botte, Universität Gent (Fast Interface Bus for Multiprocessor Applications), Andreas Döring (Swiss-FORTH und LMI-Metacompiler für Siemens 80C166), Rupert Nieberle (Echtzeitsysteme für interaktive musikalische Anwendungen). Spontane Beiträge hörten wir schließlich von Brühl über die Zukunft von Forth, ein Vortrag mit Witz und Mut und den Ausblick "Forthprozessoren bald selbst zu machen", sowie von Klingelnberg zum FPC und dessen Gebrauch. Der Abend ließ dann wiederum genügend Raum für die Lobby.

Neben den Vorträgen gab es in der Ausstellung allerlei Interessantes zu bestaunen. Die Attraktion war natürlich die Musikmaschine von CAMP und Rupert Nieberle in Aktion. Sein System begleitete dann auch mit einer eigenartigen sanften eigentümlichen rhythmischen, mir etwas fremden aber doch irgendwo ansprechenden Musik das Abendessen. Gezeigt wurden zum einen verblüffend einfache Applikationen wie das LED-Modell zur Darstellung der Ozonmessung mittels Laserstrahl-Rückstreuung, Forth im 65F12, Dr. Hegewisch, Uni München; dann aber auch lei-

stungsstarke Geräte wie z.B. das MC-32 Meßsystem realisiert mit Z80/RTX, Klaus Kohl. Ihr Können zeigten ferner Catrin Kilian mit tiny-Forth, Flesch Forth-Systeme, Bernd Paysan, Ing. Büro Flögel, Brühl Elektronik, Jörg Plewe und nochmal Klaus Kohl/Heinz Schnitter mit dem Forth im Zilog Super8 - ich hoffe ich habe niemanden vergessen.

Der Sonntag schließlich gehörte wie immer dem Verein. Die Vereinsversammlung war dieses Jahr ausgezeichnet besucht. Der Verlauf war sehr turbulent, streckenweise sogar dramatisch, weil dieses Jahr wieder das Direktorium gewählt werden mußte und leider Heinz Schnitter nicht wieder kandidierte. Und ohne ihn auch nicht das übrige Direktorium. Neu gewählt wurden schließlich Hartmut Pfüller und Gert-Ulrich Vack und bestätigt wurde Christoph Krinninger. Das Forth Büro jedoch mit der Mitgliederverwaltung blieb bei Ulrike Schnitter in München unverändert bestehen. Unverändert blieben auch die Mailbox beim neuen Sysop Jens Wilke sowie der Forth-Vertrieb der PD-Versionen bei Johannes Teich und die Redaktion der Vierten Dimension.

**Autor: Michael Kalus**

# Die neue Münchner FORTH-Box

## Eine Einführung

FORTH-Gesellschaft e.V. Postfach 1110 W-8044 Unterschleissheim

Netzwerkadresse: BBS.FORTH-ev.de 089/8714548 300-2400 Baud 8n1

Falls Sie User dieser Box werden moechten, bitte jetzt NEW eingeben!  
Wenn Sie nur mal reinschnuppern wollen, geben Sie GAST oder GUEST ein!

Username or NEW: test

Password:

Logging in: test, \*2237

Previous login: 00-Jan-00 (22 new messages)

Last caller: jaw (Jens Wilke SYSOP)

UUCP-Knoten der FORTH-Gesellschaft e.V., Unterschleissheim

BBS\_FORTHEV            D-8000 Muenchen            Streitbergstr.73a  
110 .. 2400 bps / 8n1 / 24 h / 089-8714548 / ++49-089-8714548

FORTH gateway: Usenet/Eunet --> Zerberus  
Net addresses: forthev.UUCP, BBS.FORTH-ev.de

1 mail message. (1 new)

[Local Board Forum]

9:18a (?=help!) -> feed

Feeeeeeeeeeeeeeeeeeeeedback, man.  
Eine Nachricht beim Sysop abladen?

(y/n) y

Subject: Hallo

Type Letter; single / when done.  
/s to save message or /q to quit.

- 1> Hallo lieber Sysop!
- 2>
- 3> Ich schreibe Dir nur mal schnell
- 4> eine kleine Nachricht damit die
- 5> VD Leser sehen, wie man bequem
- 6> eine Mitteilung an den Sysop
- 7> schreiben kann.
- 8> tschuess
- 9> /s

Letter saved.

[Local Board Forum]

9:23a (?=help!) -> mail

1 24 May 91 - jaw (Jens Wilke SYSOP)

Next/1 Mail> 1

To: test

Subject: Hallo

From: jaw (Jens Wilke SYSOP)

Comments: --

Date: Fri, 24 May 91 09:22:47 MSZ

Hallo!

Das ist eine kleine Nachricht fuer  
unseren TEST-Account.

-----

Reply to:jaw@BBS.FORTH-ev.de  
(Jens Wilke SYSOP) The BBS of  
FORTH-ev...

(DONE) Mail> list

1 24 May 91 - jaw (Jens Wilke SYSOP)

(DONE) Mail> ?

AGAIN        - Nachricht nochmal anzeigen  
BACK         - letzte Nachricht  
CLEAR        - Bildschirm loeschen  
DELETE       - Nachricht loeschen  
EXPAND       - Zeigt eine Empfaengerliste  
FOLLOW       - Antwort senden mit >Quotes  
FORWARD      - Leitet Nachricht um

# Die neue Münchner FORTH-Box

## Eine Einführung

HELP	- Zeigt auch diesen Text an	
LIST	- Listet die Nachrichten	Post, List, [1-4] Quit: li 1 date
NEXT	- Liest naechste Nachricht	
QUIT	- Verlaesst den Bereich	001: 06 Sep 90 22:22 GMT
REPLY	- Sendet Nachricht an den Absender	002: 06 Sep 90 22:23 GMT
SEND	- Sendet eine neue Nachricht	003: 06 Sep 90 22:24 GMT
VERBOSE	- wie AGAIN mit vollem Kopf	004: 06 Sep 90 22:25 GMT
WRITE	- Schreibt Nachricht in ein File	Post, List, [1-4] Quit: quit
(DONE) Mail> qui		[Local Board Infos]
Purge all letters? n		9:28a (?=help!) -> join
[Newsgroup Forum]		JOIN / NEW E i n s t e l l u n g
9:25a (?=help!) -> forums		Hier koennen Sie Newsgroups anwaehlen die beim Kommando NEW nach neuen Nachrichten durchsucht werden. Mit FORUMS bekommen Sie eine Uebersicht der vorhandenen Newsgroups.
Die lokalen Bretter:		? gibt (wie ueberall) Hilfestellung.
INFOS hier das Neueste von der/über die FORTH-Gesellschaft e.V.		Configure your NEW scan? y
FORUM Diskussionen		Currently JOINED to newsgroups:
USER Vorstellung der User (Bitte jeder mal reinschauen!)		(0 new) local.infos
Die internationalen Bretter:		(2 new) local.forum
COMP.LANG.FORTH Diskussionen ueber FORTH		(0 new) local.user
ALT.BBS.WAFFLE Diskussionen ueber das Mailboxprogramm WAFFLE		(0 new) monitor
Die Bretter koennen Sie einfach anwaehlen, indem Sie den entsprechenden Namen eingeben. Um die Nachrichten im Brett zu lesen, geben Sie READ ein. POST für Nachricht ins Brett.		JOIN / NEW E i n s t e l l u n g
[Local Board Forum]		Hier koennen Sie Newsgroups anwaehlen die beim Kommando NEW nach neuen Nachrichten durchsucht werden. Mit FORUMS bekommen Sie eine Uebersicht der vorhandenen Newsgroups.
9:27a (?=help!) -> infos		? gibt (wie ueberall) Hilfestellung.
[Local Board Infos]		Join command ADD DELETE FORUMS LIST QUIT? li
9:27a (?=help!) -> read		Currently JOINED to newsgroups:
[Local Board Infos]		(0 new) local.infos
(No new messages / highest is 4)		(2 new) local.forum
Post, List, [1-4] Quit: list 1		(0 new) local.user
001: Die Programmierumgebung FORTH		(0 new) monitor
002: Die FORTH-Gesellschaft e.V.		Join command ADD DELETE FORUMS LIST QUIT? forums
003: Die FORTHEV-Bretter		
004: Mitgliedschaft		



# Die neue Münchner FORTH-Box

## Eine Einführung

```
Die lokalen Bretter:
INFOF hier das Neueste von der/über
      die FORTH-Gesellschaft e.V.
FORUM Diskussionen
USER  Vorstellung der User (Bitte
      jeder mal reinschauen!)

Die internationalen Bretter:
COMP.LANG.FORTH
      Diskussionen ueber FORTH
ALT.BBS.WAFFLE
      Diskussionen ueber das
      Mailboxprogramm WAFFLE

Die Bretter koennen Sie einfach anwaeh-
len, indem Sie den entsprechenden Namen
eingeben. Um die Nachrichten im Brett
zu lesen, geben Sie READ ein. Wenn Sie
eine Nachricht ins Brett abschicken mo-
echten, geben Sie POST ein.

Join command ADD DELETE FORUMS LIST
QUIT? del infos

Currently joined. UNJOIN infos ? y

Join command ADD DELETE FORUMS LIST
QUIT? li

Currently JOINED to newsgroups:
[Local Board Infos]
9:38a (?=help!) -> files

Protokolle mit P waehlen:

X Xmodem
Z Zmodem
P Puma
Y Ymodem
K Kermit
? zeigt Befehlsuebersicht
  Nur einen Buchstaben tippen

(2 new) local.forum
(0 new) local.user
(0 new) monitor

Join command ADD DELETE FORUMS LIST
QUIT? qui

(>Protocol: ?

Uebersicht:
K - Kermit
M - MPt (Puma!)
X - Xmodem
Y - Batch Ymodem
Z - Zmodem

Protokoll Z wird benutzt.

(>Protocol: Z
Z protocol selected

(>

Die Oeffentlichen Directories:

/DOCS      Informationen und WAFFLE
           Bedienungsanleitung
/SOURCE    Sourcecode fuer alle
           Systeme
/MSDOS     Programme fuer MS-DOS
/ATARI     Programme fuer ATARI ST
/AMIGA     Programme fuer den
           Commodore AMIGA

(>Log
Was: /public
New: docs

(>Directory:

user.doc      49356      03-Jan-91
              Die Anleitung fuer die User

(>Send: user.doc
386 blocks, 49k

>Uebertragung laeuft<

(>Quit..

[Local Board Infos]
9:40a (?=help!) -> files

Protokolle mit P waehlen:

X Xmodem
Z Zmodem
```

## Die neue Münchner FORTH-Box Eine Einführung

P Puma		N)EW	- Sucht nach neuen Files seit einem Datum
Y Ymodem		P)ROTOCOL	- Waehlt ein Protokoll aus
K Kermit		Q)UIT	- Verlaesst den Filebereich
? zeigt Befehlsuebersicht		R)ECEIVE	- Empfaengt ein File (Danke!), Z Protokoll
Nur einen Buchstaben tippen!		S)END	- Sendet ein File (fuer Sie!), Z Protokoll
(>		V)IEW	- Zeigt ein ASCII File an, mit [more]
[ ] the FILES menu..		.	- Springt ein Directory zurueck
A)RC	- Inhalt von .ARC .ZOO .ZIP ... Archiven	*	- Zeigt alle anwaehlbaren Directories
C)OPY	- Eingeben eines ASCII Files	(>Quit..	
D)IR	- Anzeige der vorhandenen Files	[Local Board Infos]	
F)IND	- Nach Filenamen suchen	9:40a (?=help!) -> bye	
H)ELP	- Hilfe	test logged out 24-May-91 9:40a	
L)OG	- Wechselt das Directory		
M)ASTER	- Master File Directory		

*Im folgenden möchte ich eine kurze Einführung in die Bedienung des Mailboxprogramms geben und auf nützliche Dinge hinweisen die normalerweise unentdeckt bleiben..!*

### Stichwörter: Mailbox, DFÜ, Waffle

Die Münchner FORTH-Box ist zu erreichen unter: Tel.089/8714548 rund um die Uhr  
300-2400 Baud; Parameter:8N1

Datenfernübertragung (DFÜ) kann man fast schon zu den Modewörtern zählen wie z.B. objektorientiertes Programmieren oder Fuzzy Logic. Die DFÜ erlebt momentan wieder einen großen Aufschwung. Endlich sind Modems mit hohen Übertragungsgeschwindigkeiten für jeden erschwinglich geworden. So ist es fast jedem engagierten Computerfreak heute möglich seine eigene Mailbox zu betreiben. Wer die Artikel aus den letzten VD's verpaßt hat hier nocheinmal kurz das Wesentliche:

Auch wir sind natürlich mit auf die DFÜ-Welle aufgesprungen und das aus gutem Grund, so ist es doch

besonders für die in alle Himmelsrichtungen versprengten FORTH'ler wichtig miteinander Kontakt aufzunehmen. Daß 'normale' Post dazu nicht immer den besten Weg darstellt ist klar. Erfahrungsaustausch oder richtige Diskussionen sind auf diesem Weg fast ausgeschlossen.

Das Netzwerk, mit dem wir verbunden sind, ist das weltweit größte und heißt Internet. Im Internet gibt es sog. Newsgroups in denen man FORTH Nachrichten und Erfahrungen austauschen kann. Die für uns natürlich besonders interessante Newsgroup ist COMP.LANG.FORTH, das steht für COMPUTER, LANGUAGES und das FORTH am Schluß braucht wohl nicht näher erläutert zu werden. In diesem Newsgroup wurde in letzter Zeit hauptsächlich über den neuen ANS-Standard diskutiert und sogar Vorschläge eingereicht. WELTWEIT!

Aber nun zu etwas mehr 'Handfestem' sprich: Zur Praxis!

Im Folgenden möchte ich eine kurze Einleitung in die Funktionen und Bedienung unseres Mailboxprogramms geben. Anbei ist auch noch ein Online-Mitschnitt zu finden, in dem man die Reaktionen des Mailboxsystems sehen kann.

Wenn die erste Hürde überwunden ist und Sie Ihr System dazu überredet haben, die Zeichen unserer Begrüßungsmeldung korrekt darzustellen (8 Bit, No Parity, 1 Stop Bit), kann es nur noch aufwärts gehen. Falls sie an dieser Stelle schon Probleme haben sollten und erst keine richtige Verbindung zustande kommt, stehe ich für Fragen zur Verfügung (Adresse am Ende des Textes).

Wie man sich in das System einloggt ist eigentlich selbsterklärend, also steigen wir gleich mitten in die Befehle des Mailboxprogramms ein.

Waffle, unser Mailboxprogramm, ist befehlszeilenorientiert, d.h. daß man auf schöne Menüs, wie man es

# Die neue Münchner FORTH-Box

## Eine Einführung

von den ganzen Fido-Boxen her kennt, leider verzichten muß. Die Befehle können übrigens auch abgekürzt werden.

Generell bekommt man mit '?' eine kleine Übersicht der Befehle, wenn man gar nicht mehr weiter weiß.

### **Mails verschicken und empfangen:**

Der zentrale Befehl für die persönlichen Nachrichten ist MAIL. Nach dem eingeben des Befehls MAIL befindet man sich im MAIL-Bereich. Das bedeutet, hier haben andere Befehle ihre Gültigkeit (siehe Mitschnitt). Hier ein paar Beispiele der Befehle:

#### **Mail> SEND SYSOP**

Sendet dem Sysop einen Brief Oder man gibt z.B. direkt MAIL SY-SOP ein.

#### **Mail> LIST**

Listet ihre erhaltenen Briefe ab Nr.3

#### **Mail> SEND peter@laiv.selbig.de**

Versendet eine Nachricht über das Netzwerk

### **Diskussionen**

#### **und Nachrichtenbretter:**

Man kann natürlich nicht nur private Nachrichten verschicken sondern auch öffentliche. Die öffentlichen Nachrichten werden in FORTH nach Themen sortiert abgelegt. Um einen Newsgroup anzuwählen gibt man einfach dessen Namen ein. Jetzt kann man mit dem Befehl READ Nachrichten ansehen oder mit dem Befehl POST Nachrichten abschicken. Eine Übersicht über die anwählbaren Bretter erhält man übrigens mit FORUMS.

Aber nun wollen wir den Befehl READ mal etwas näher beleuchten:

Nach dem Eingeben von READ haben wieder nur spezielle Kommandos ihre Gültigkeit.

Das wohl wichtigste Kommando ist LIST. Mit ihm bekommt man eine Übersicht der vorhandenen Artikel bzw. eine Aufzählung derer Subjects. LIST fängt immer an der aktuellen Nachricht an, es sei denn, Sie geben eine andere Nummer ein. Statt Subject kann man auch andere Dinge, die im Kopf der Nachricht stehen, anzeigen. Hier ist zum Beispiel DATE oder FROM sinnvoll. Um es ein wenig klarer zu machen noch ein paar Beispiele:

#### **LIST<**

Listet die Subjects aller Artikel

#### **LIST DATE<**

Listet das Datum von jedem Artikel

#### **LIST 1600 DATE<**

Listet die Erstellungsdaten ab Art. 1600

Okay, falls es immer noch nicht ganz klar sein sollte, ist ja noch das abgedruckte Protokoll vorhanden.

LIST hilft ja noch nicht sehr viel, denn eigentlich möchten Sie ja die Nachrichten lesen. Kein Problem! Drücken Sie einfach mal RETURN und schon kommt die nächste Nachricht. Wenn Sie sich nicht alles gemerkt haben, geben Sie einfach AGAIN ein und schon wird die Nachricht nochmals angezeigt. Um eine Nachricht direkt anzuwählen brauchen Sie nur die Nummer einzugeben.

Bis jetzt war es ja noch recht einfach, aber jetzt wollen wir mal sehen was so alles im Mailboxprogramm drinsteckt.

### **News lesen**

JOIN ist ein Befehl mit dem man die FORTH anwählen kann, die man gerne regelmäßig lesen möchte. Der Computer speichert nun für Sie welche Nachrichten Sie schon gelesen haben und zeigt Ihnen nur die Neuen. Um die neuen Nachrichten zu lesen gibt man den Befehl NEW ein.

Seine Bedienung ist identisch mit der von READ. Sie können aber natürlich später immer noch eine Nachricht lesen die schon längst abgehakt ist, falls sie sich noch im System befindet.

Falls Sie Telefonkosten sparen möchten gibt es noch eine elegante Möglichkeit: Ich habe für solche Fälle extra ein Programm eingerichtet. Nachdem man POLL eingibt sendet das System die neuesten Nachrichten gepackt und mit dem gewünschten Protokoll.

Nun kann man bequem offline die Nachrichten wieder auspacken und anschauen. Falls Sie nicht zum Orts-tarif anrufen, sollten Sie auf jeden Fall von diesem Service Gebrauch machen!!!

Wenn Sie es benutzen wollen, brauchen Sie nur eine kurze Mail an mich (JAW bzw. SYSOP) zu schreiben, in der steht, welche Bretter Sie gerne auf diesem Weg beziehen möchten. Damit ich gleich weiß was gemeint ist, bitte als Subject POLL-TOOL eingeben.

### **Up- und Downloads**

Bevor ich den File-Bereich beschreibe (darauf warten bestimmt schon einige) möchte ich kurz noch erwähnen, daß man Nachrichten auch per Übertragungsprotokoll ins System senden kann, statt diese mühsam per Hand einzugeben. Wenn der Editor die erste Zeile verlangt gibt man '/' und RETURN ein (damit gelangt man in die Befehls-ebene des Editors; '?' für eine Übersicht) und gibt dann UPLOAD ein. Das System verwendet das von Ihnen im Filebereich gewählte Protokoll (Erklärung folgt). ACHTUNG: Vergewissern Sie sich immer, ob das richtige Protokoll angewählt ist, bevor Sie eine Übertragung starten! Der Computer merkt sich zwar Ihr "persönliches Protokoll", falls aber das falsche angewählt ist, kann das unerwartete Folgen haben.

# Die neue Münchner FORTH-Box

## Eine Einführung

### Der FILE-Bereich

Hier findet man all das, was des FORTH'lers Herz höher schlagen läßt. Jede Menge Source-Code aber auch viele nützliche Utilities und sogar komplette Forth-Systeme.

Mit dem Befehl FILES taucht man in die Tiefen des File-Bereichs ein. Mit diesen Befehlen kann man nun in den öffentlichen Directories der Mailbox herumstöbern. Sie brauchen nur den Anfangsbuchstaben einzugeben, um den Befehl anzuwählen. Mit "\*" bekommen Sie eine Übersicht der vorhandenen Directories. Mit dem Befehl "L" (wie LOG) können Sie das Directory anwählen (funktioniert so wie CD!). Mit "D" (wie DIR) erhalten Sie eine Übersicht der Files, die sich im aktuellen Directory befinden. Und nun das wohl Wichtigste: Mit "S" überträgt Ihnen das System ein File. Aber Vorsicht! Zuvor müssen Sie noch mit dem Befehl P (wie Protocol) ein Übertragungsprotokoll auswählen. Mit "P?" erhalten Sie eine Übersicht der verfügbaren Protokolle.

Der Befehl A (ARC) ist auch noch bemerkenswert. Mit ihm kann man den Inhalt eines Archivfiles anzeigen lassen. Das funktioniert mit den Pakkern: ARC, ARJ, LZH, SIT, TAR, ZIP, ZOO.

### Ein paar Tips

Jeder User hat im Mailboxrechner sein persönliches Directory, in dem seine Post gespeichert wird.

Dieses Directory kann man übrigens auch im FILE-Bereich anwählen, es hat den Namen /USER/ plus Ihrem Usernamen.

Damit kann man einige nützliche Dinge anstellen. So kann man innerhalb von READ eine Nachricht mit dem Befehl WRITE in ein File in seinem persönlichen Directory ablegen. Falls kein Filename angegeben wurde, wird die Nachricht an das File MESSAGES angehängt.

Das Gleiche kann man auch innerhalb von MAIL machen, mit dem Unterschied, daß standardmäßig in das File MAILBOX geschrieben wird. Das kann man gut verwenden, um einige interessante Messages zu sammeln und diese dann hinterher im FILE-Bereich downloaden.

Im privaten Directory können auch sog. Signature-Files stehen. Diese Files werden an jede Nachricht angehängt, die Sie schreiben. Hier kann man z.B. seine Netzwerkadresse schreiben oder seine richtige Adresse, je nach belieben. Es gibt drei verschiedene Signature-Files: MAILSIG wird an private Nachrichten angehängt, LOCALSIG wird an öffentliche Nachrichten angehängt, die innerhalb der Box bleiben, INTERSIG wird an internationale öffentliche Nachrichten angehängt.

Mit dem Befehl "EDIT filename" kann man beliebige Files in seinem privaten Directory erzeugen und editieren. Natürlich kann man das auch bequem daheim machen und sie dann einfach im FILE-Bereich uploaden.

Wer viele internationale Briefe schreibt findet das File ALIASES bestimmt auch ganz nützlich. Darin kann man Kürzel für längere Adressen angeben. Das Format ist denkbar einfach: Links steht das Kürzel und rechts die entsprechende Adresse.

In einer anderen Mailbox könnte das so aussehen:

**jens jaw@BBS.FORTH-eV.de usw.**

Wer mehr über das Netzwerk und die Adressierung wissen möchte, kann sich mal die Hilfetexte in HELP NET anschauen.

Die Tips habe ich absichtlich etwas genauer behandelt, da man diese Hinweise normalerweise erst noch langem Suchen in der Bedienungsanleitung entdeckt.

Die restlichen Befehle der Mailbox und ihre genaue Anwendung kann man der ausführlichen Bedienungsanleitung entnehmen. Sie ist im FILE-Bereich für jedermann erhältlich.

In der Mailbox sind übrigens auch noch jede Menge Tips zu finden. Mit dem Befehl HELP bekommt man auch nützliche Informationen.

Wer Probleme hat mit der Mailbox Kontakt aufzunehmen kann mich gerne anrufen oder schreiben:

**Jens Wilke**  
Streitbergstr.73a  
8000 München 60  
089/8714352

```

\ A Taste of...

: " ( -- ) Ascii " word drop ; immediate
" A Taste of SMALTALK "

: ( ( -- ) Ascii ) word drop ; immediate
( A Taste of PASCAL )

: COMMENT ( -- ) Ascii ; word drop ; immediate
COMMENT A Taste of ALGOL ;

: Comment: ( -- )
  Create bl word 1+ c@ ,
Does> ( -- ) dup c@ swap 1+ c@ swap
  BEGIN
    over word count
    ?dup 0= IF 2drop exit Then
    1- + c@ over =
  UNTIL 2drop ;

Comment: (* *)
(* A Taste of MODULA & ANOTHER Taste of PASCAL *)

Comment: /* */
/* A Taste of C, PROLOG and PL/1 */

: # ( -- ) [compile] \ ; immediate
# A Taste of TeX

: REM ( n -- ) drop [compile] \ ; immediate
10 REM A Taste of BASIC

: -- ( -- ) [compile] \ ; immediate
-- A Taste of ADA

: C ( -- ) [compile] \ ; immediate
C      A Taste of FORTRAN

: ; ( -- )
[compile] \ ; immediate
; A Taste of Assembler and LISP

\A Taste of ...FORTH-Tagung '91
U. Hoffmann

```

# 7th euroFORML conference

## Call for papers

### *7th euroFORML conference on the FORTH programming language and FORTH processors*

Held from October 11th through 13rd 1991  
at the Casino of Marianske Lazne (Marienbad)  
Czechoslovakia In cooperation with Forth Interest Group  
Inc, USA.

euroFORML is an international meeting of computer practitioners using FORTH as a problem solving tool. Lectures, workshops and presentations will be conducted to demonstrate techniques and problem solving strategies that have proved useful.

This years conference will again focus on FORTH in real time applications and its potential in the area of robotics and industrial control applications. At euroFORML'89 for the first time participants from East-European countries participated and presented some very interesting work. FORTH is being used in Leningrad (USSR), Tartu (Estonia), Sofia (Bulgaria) as well as in former East-Germany and other East-European countries.

Given the current political circumstances and the non-convertibility of their currencies participation for East-Europeans became even more difficult than two years ago. This is why we decided to hold this years conference at a central location in Eastern-Europe. Marianske Lazne is located mid-way between Nuremberg and Prague. As the worlds first spa it has always been an international meeting place besides accomodating the European nobility. The hotels certainly do not live up to western standards which all too often destroys the atmosphere.

Marianske Lazne has plenty of it. Accomodation will be in a hotel with single and twin bedrooms right next to the Casino were the meetings will be held. Organising the meeting from Hamburg under the particular economic conditions of the CSFR will be more involved then usual.

Therefore we ask you to register early. The conference language will be English and FORTH, of course. The conference will be self organizingie. There will not be a strict agenda prior to the conference. If you want to present your ideas you may choose one of the following formats:

#### *Paper presentation*

You will present a paper in a 15 minute talk in front of the whole group with the possibility to get immediate feedback.

#### *Poster presentation*

You will be assigned a "poster space" where you can present your ideas to a small group of people in a separate room. This is especially useful for demonstrations of hard- and software.

#### *Workshops*

You may organize or participate in workshops which will be organized at the beginning of the conference, depending on demand.

An English language proceedings will be published after the conference; papers to be included in the proceedings will be handed out at the beginning of the conference. Registrations should be mailed by July 1st in order to be eligible for a rate reduction. A deposit per person will be required. There is a limited number of places available for students at a substantially reduced rate. A limited number of participants from East-Europe may register till July 1st at a special rate payable in Czechoslovakian currency. Space is limited and you are assigned on a first-come, first-serve basis. More specific information about the rates will be mailed in the next couple of weeks.

#### *Author instructions*

Papers to be presented at the conference (to be included in the proceedings) will have to be mailed to the organizer no later then September 23rd 1991 in camera ready form.

The format is DIN A 4 (letter size) with a margin of 2,5 cm (1.5") on all sides. Every page should have a page number and the authors name. Papers should not exceed 15 pages. Code should be accompanied by shadow screens.

**For reservations and conference papers write or call  
euroFORML - office**

**M.Kern, K.Schleisiek-Kern  
Uhlenhorster Weg 3 / W-2000 Hamburg 76 /  
Germany Tel: +49 40 2296441 Fax: +49 40 2297205**

# --FORTH-Gruppen--

## **FORTH-Gruppen:**

W-1000 Berlin  
Claus Vogt  
Tel. 030/2 16 89 38  
Treffen nach Absprache

W-4130 Moers 1  
Friederich Prinz  
Tel. 02841/5 83 98  
Treffen nach Absprache

Gruppe Rhein-Ruhr:  
Jörg Plewe  
Tel. 0208/42 35 14  
W-4000 Düsseldorf  
Gebäude des S-Bahnhof  
Derendorf,  
Münsterstraße 199  
Treffen jeden ersten  
Sonnabend im Monat

W-6800 Mannheim  
Thomas Prinz  
Tel. 06271/28 30  
Ewald Rieger  
Tel. 06239/86 32  
Treffen jeden ersten  
Mittwoch im Monat im  
Vereinslokal des Segel-  
vereins Mannheim e.V.  
Flugplatz, Mannheim-  
Neuostheim

W-7000 Stuttgart  
Wolf-Helge Neumann  
Tel. 0711/88 26 38  
Treffen nach Absprache

O-Leipzig  
FORTH-Gruppe Leipzig:  
Michael Balig, Lützner  
Plan 17  
O-7033 Leipzig  
Dr. Jürgen Hesse  
Tel. 041/69 56 02  
Liselotte-Herrman-Str. 40  
O-7050 Leipzig

## **FORTH für Ratsuchende:**

Jörg Staben  
Tel. 02103/5 56 09  
dienstags und freitags  
von 20.00-22.00 Uhr

Frank Stüss  
Tel. 06187/9 15 03

Karl Schroer  
Tel. 02845/2 89 51

Andreas Findewirth  
Tel. 05221/2 35 04

Andreas Jennen  
W-1000 Berlin, UUCP

Jörg Plewe  
Tel. 0208/42 35 14

## **FORTH Fachgruppen:**

W-6800 Mannheim  
FIS (FORTH Integriertes  
System) - Datenbank,  
Textverarbeitung,  
Kalkulation  
Postadresse:  
Dr. med.  
Elemer Teshmar  
Danziger Baumgang 97  
W-6800 Mannheim 31

## **FORTH Interessengebiete:**

volksFORTH/ultraFORTH  
Klaus Kohl  
Tel. 08233/3 05 24  
Klaus Schleisiek-Kern  
Tel. 040/2 20 25 39

32-BIT Systeme  
Robert Jones  
Tel. 02434/45 79

Künstliche Intelligenz  
Ulrich Hoffmann  
Tel. 0431/67 88 50

NC4000 Novix Chip  
Klaus Schleisiek-Kern  
Tel. 040/2 20 25 39

Relationale Netze  
Künstliche Intelligenz  
Realtime  
Wigand Gawenda  
Tel. 040/44 69 41

Gleitkomma-Arithmetik  
Andreas Döring  
Tel. 0721/59 39 35

32FORTH  
Rainer Aumiller  
Tel. 089/6 70 83 55

PostScript/FORTHscript  
Christoph Krininger  
Tel. 089/ 7 25 93 82

FORTH im Unterricht  
Rolf Kretschmar  
Tel. 0240/43 90

Objekt-orientiertes FORTH  
Christoph Krininger  
Tel. 089/7 25 93 82  
Ulrich Hoffmann  
Tel. 0431/67 88 50

F-PC Zimmer FORTH  
ASYST Meßtechnik  
Arndt Klingelberg  
Tel. 02404/6 16 48  
box:geo1:klingsberg

## **FORTH Fachgruppengründung**

Grafik, Arithmetik  
W-7000 Stuttgart 80  
Jörg Tomes  
Tel. 07 11/7 80 22 93  
nur am Wochenende

## **FORTH Gruppengründung:**

W-3300 Braunschweig  
Martin Holzapfel  
Tel. 05 31/35 12 62  
32-Bit Systeme

## **FORTH-MAILBOX**

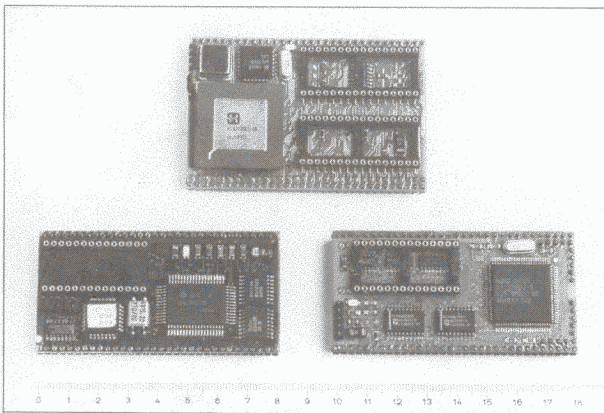


Jens Wilke (SYSOP)  
Tel. 089/8 71 45 48  
300-2400 baud  
Parameter 8 N 1

### 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

### ModuNORM



CPU-Steck-Module im Scheckkartenformat:

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

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

### LMI 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   | ● 8096/97     |
| ● Z80         | ● HD64180     |
| ● 8080/8085   | ● 8031/32/535 |
| ● 68000       | ● 6303        |
| ● Z8          | ● 6502        |
| ● 1802        | ● V25         |
| ● 6809        | ● 68HC11      |
| ● 65816/65802 | ● RTX 2000    |

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

Forth Programme, die mit einem LMI interaktiven Forth System z. B. PC/FORTH oder Z80 Forth geschrieben und getestet wurden, werden im Normalfall mit nur geringen Änderungen übersetzt.

### Serieller ROM/RAM Simulator

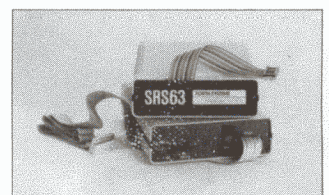
Entwickeln Sie romfähige Programme ?

Müssen Sie neu entwickelte Einplatinencomputer testen ?

Setzen Sie 2764, 27128, 27256, 27512 oder 4364, 43256 oder kompatible ROM/RAM-Bausteine ein ?

Wollen Sie diese Bausteine mit bis zu 38 400 Baud über die serielle Schnittstelle laden ?

Können Sie eine zusätzliche serielle Schnittstelle über den Speichersockel zum interaktiven Programmieren gebrauchen ?



**Dann ist unser SRS63 die optimale Ergänzung Ihres Arbeitsplatzes.**

Sie werden vom Preis-Leistungsverhältnis überrascht sein.

Unsere ROM-Compiler liefern direkt verwendbare Dateien, wir akzeptieren auch Intel-Hex oder Motorola-S-Formate.

