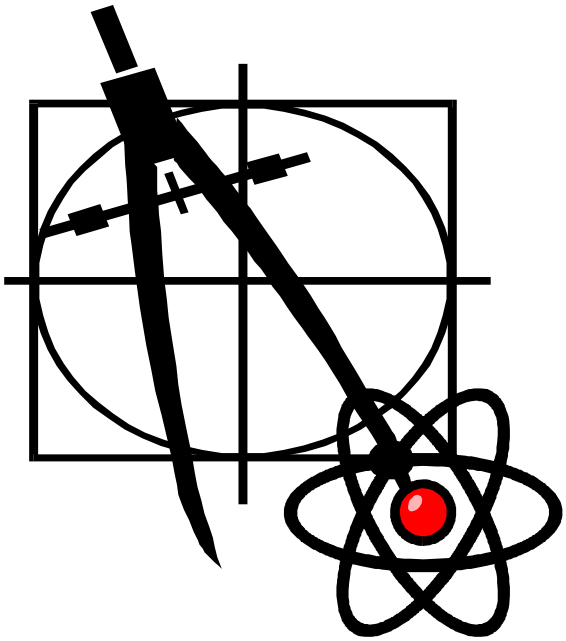
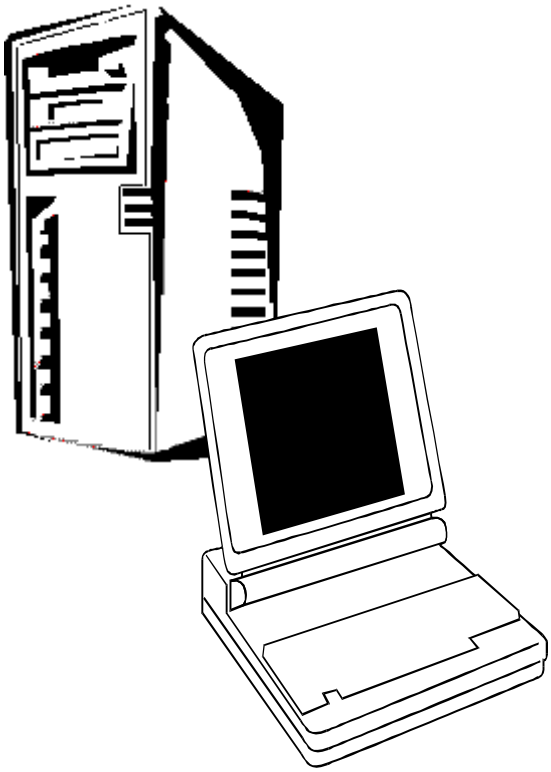
VIERTE DIMENSION

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





In dieser Ausgabe:

14. Jahrgang ; 10,- DM

Organ der FORTH Gesellschaft e.V.

Dienstleistungen und Produkte fördernder Mitglieder des Vereins

FORTH - Shirt

Räumungsverkauf

T - Shirt: hellgrau / grün
in Größe M-L-XL **15 DM**
Sweat-Shirt: grau / grün
in Größe M-L-XL **25 DM**
(+ Porto)

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

In eigener Sache

Das „Editoriat“ der VD wird geleitet von: Friederich Prinz,
Hombergerstraße 335
47443 Moers
Tel.: 02841-58398
F.PRINZ@MHB.GUN.DE
FPRINZ@T-ONLINE.DE
VD@FORTH-EV.DE

Alle Anfragen bitten wir über diese Adressen abzuwickeln, bzw. alle Beiträge an eine dieser Adressen zu senden.

Ihre Beiträge bitten wir entweder in „plain ASCII“ einzureichen, oder in einem Format, das von WinWord 6.x aufgenommen und konvertiert werden kann. Sollten Ihre Beiträge Bilder (Fotos, Skizzen, andere Darstellungen) enthalten, bitten wir zu beachten, daß die Qualität dieser Bilder durch die Wiedergabe in Graustufen und die anschließende Vervielfältigung erheblich leiden kann. Bilder, Skizzen u.ä. sollten Ihren Beiträgen separat in einem gängigen Grafikformat beigefügt sein.

Ihre Anzeige...

...könnte hier stehen - wenn Sie förderndes Mitglied der Forthgesellschaft sind und sich hier bekannt machen möchten. Bitte schreiben Sie uns !

KIMA Echtzeitsysteme GmbH

Tel.: 02461/690-380
Fax: 02461/690-387 oder -100
Karl-Heinz-Beckurtz-Str. 13
52428 Jülich

Automatisierungstechnik: Fortgeschrittene Steuerungen für die Verfahrenstechnik, Schaltanlagenbau, Projektierung, Sensorik, Maschinenüberwachungen. Echtzeitrechnersysteme: für Werkzeug- und Sondermaschinen, Fuzzy Logic

FORTech Software GmbH

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

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

Ingenieurbüro Dipl.-Ing. Wolfgang Allinger

Tel.: (+Fax) 0+212-66811
Brander Weg 6
D-42699 Solingen

Entwicklung von µC, HW+SW, Embedded Controller, Echtzeitsysteme 1-60 Computer, Forth+Assembler PC / 8031 / 80C166 / RTX 2000 / Z80 ... für extreme Einsatzbedingungen in Walzwerken, KKW, Medizin, Verkehr / >20 Jahre Erfahrung.

Ingenieurbüro Klaus Kohl

Tel.: 08233-30 524 Fax: —9971
Postfach 1173
D-86404 Mering

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

Editorial

IMPRESSUM

Name der Zeitschrift

Vierte Dimension
Organ der Forth-Gesellschaft e.V.

Herausgeberin

Forth-Gesellschaft e.V.
Postfach 1110
85701 Unterschleißheim
Tel./Fax: 089-317 37 84
E-Mail:
SECRETARY@ADMIN.FORTH-EV.DE

Redaktion & Layout

Friederich Prinz
Hombergerstraße 335
47443 Moers
Tel.: 02841-58 3 98
E-Mail:
F.PRINZ@MHB.GUN.DE
FPRINZ@T-ONLINE.DE

Anzeigenverwaltung
Büro der Herausgeberin

Redaktionsschluß 1997
März, Juni, September, Dezember
jeweils in der letzten Woche

Erscheinungsweise
1 Ausgabe / Quartal

Einzelpreis
DM 10,- zzgl. Porto u. Verp.

Manuskripte und Rechte

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

Liebe Leser,

es tut sich was - in und mit der VD ebenso wie in der FG. Dem Zwang zum Sparen konnte sich auch die Forthgesellschaft nicht länger entziehen. Um die Quartalsschrift des Vereins dauerhaft zu sichern und gleichzeitig lange überfällige Aktivitäten zu ent-wickeln und auszuweiten, war das Direktorium gezwungen, sowohl die VD, als auch Strukturen innerhalb des Vereins den schwindenden Mitgliederzahlen und dem sich damit verringernden Beitragsaufkommen anzupassen (siehe Bericht des Direktoriums). Für die VD, die ab dieser Ausgabe ausschließlich auf „ ehrenamtlichen Beinen“ steht, heißt dies konkret, auf Liebgewonnenes zu verzichten. So werden wir auf nicht absehbare Zeit ohne „Buntes“ auskommen müssen. Das Editoriat (siehe Leserbrief M. Kalus) und der Druck, bzw. die Vervielfältigung der jeweiligen Ausgabe, werden voneinander getrennt sein. Das beinhaltet, bei Einbeziehung der unterschiedlichen Transportzeiten, daß der Redation zur Erstellung der VD (Inhalte und Layout) zukünftig 2 Wochen weniger Zeit zur Verfügung stehen. Und um die bescheidenen Fähigkeiten der Vervielfältigung per Kopierer nicht zu überfordern, werden wir weitestgehend auf die Aufnahme von Fotos und ähnlich komplexen Darstellungen in die VD verzichten.

Die neuen „Macher“ der VD - letztlich die Autoren der unterschiedlichen Beiträge - werden sich, ebenso wie Sie, liebe Leser, an ein leicht verändertes Layout gewöhnen müssen, und daran, daß es nicht mehr ganz so viele Zeichnungen und Bilder in der VD geben wird. Im Layout einfacher und preiswerter **soll die VD** inhaltlich auf dem erreichten, hohen Niveau bleiben, und - **mit Hilfe Ihrer Beiträge** - sogar noch **gewinnen**. Die VD soll zukünftig pünktlicher erscheinen und vor allem „sicherer“, mit 4 Ausgaben pro Jahr.

In diesem Sinne sieht das Editoriat der „neuen“ VD diese Einschnitte nicht als Belastung an, sondern als Option auf die Zukunft unserer Zeitschrift. Diese Zukunft erscheint heute für einen absehbaren Zeitraum gesichert. Um diese Zukunft gestalten und ausfüllen zu können, bedarf es aber nach wie vor Ihrer Mithilfe, die Sie am einfachsten durch die Bereitstellung möglichst vieler, interessanter Beiträge leisten können. Um diese Beiträge bitten wir Sie.

Für das “Editoriat”
Friederich Prinz

Bericht des Direktoriums

Dem Auftrag der Mitgliederversammlung 1997 folgend, hat das Direktorium der Forth Gesellschaft nach Wegen gesucht, auf denen trotz sinkender Mitgliederzahlen und entsprechend schwindendem Beitragsaufkommen sowohl das Erscheinen des Vereinsorgans "Vierte Dimension", als auch die anfallenden Verwaltungsarbeiten des Vereins nachhaltig finanzierbar gestaltet und gesichert werden können. Hierzu waren strukturelle Maßnahmen bezüglich der Redaktion der "VD" notwendig.

Die "VD", der Haushaltsposten, der die Vereinskasse bisher mit großem Abstand am stärksten belastet hat, wird ab dieser Ausgabe von der Moerser Forthgruppe redaktionell erarbeitet. Die Moerser Forthgruppe führt diese Arbeit für den Verein ehrenamtlich aus. In der Redaktion der "VD" eventuell anfallende Kosten werden zukünftig nur gegen Einzelnachweise durch den Verein erstattet. Hier sind im wesentlichen Kosten für Büromaterial und Porto für den Versand zur Vervielfältigung zu erwarten.

Die Vervielfältigung übernimmt ein Copyshop in Sachsen. Hier wird sich der Verein des günstigsten Angebots bedienen, das dem Direktorium für das Vervielfältigen, Schneiden, Heften und Falten der "VD" vorgelegt wurde. Organisiert wird die Vervielfältigung von Thomas Beierlein.

Damit erhält die "VD" eine völlig neue Organisationsform. Die "VD" wird nicht mehr, wie bisher, gegen Bezahlung von einem "externen" Gewerbetreibenden erstellt, sondern direkt vom Verein selbst, bzw. von seinen Mitgliedern. Lediglich die Vervielfältigung, für die der Verein nicht die erforderliche Hardware besitzt, wird an einen Gewerbetrieb vergeben.

Die hierbei zu erwartenden, erheblichen Einsparungen sollen den Mitgliedern der Gesellschaft direkt zugute kommen. Vorbehaltlich der Zustimmung durch die Mitgliederversammlung 1998 hat das Direktorium darum eine Reduzierung der Mitgliedsbeiträge beschlossen. Dies wird ein Punkt der Tagesordnung der Versammlung sein.
Die neuen Beiträge können dementsprechend erst ab dem Jahr 1999 gültig sein, so daß für das Jahr 1998 mit Überschüssen gerechnet werden kann. Diese Überschüsse sollen dazu verwendet werden, Aktivitäten zu fördern, die den Vereinszielen - formuliert in der Satzung der Forth Gesellschaft - dienen. Dies sind Aktivitäten, die der Verbreitung und Förderung von Forth dienen. In einem ersten Schritt sollen hierzu die Informationsschriften des Vereins aktualisiert werden.

Die bisher eingeleiteten Maßnahmen zur Sanierung und Konsolidierung des Haushaltes der Forth Gesellschaft, können nach Auffassung des Direktoriums nur erste Schritte auf einem als richtig erkannten Weg sein. Diesen Weg fortzusetzen und die bereits angesprochenen, sowie zukünftige Maßnahmen in diesem Sinne zu begleiten, sieht das Direktorium als seine vordringlichste Aufgabe an. Hierzu erscheint es sinnvoll, das Direktorium in seiner aktuellen Zusammensetzung zu belassen.

Aus diesem Grund wird die Tagesordnung der Jahresversammlung 1998 - vorerst - nicht die Neuwahl des Direktoriums enthalten. Sollte in der Mitgliedschaft - dem entgegen - der Wunsch nach einer Neuwahl bestehen, so kann dies auf Antrag mit Zustimmung der Versammlung als weiterer Punkt auf die Tagesordnung genommen werden.

Mit dem Wunsch, anläßlich der Jahrestagung mit möglichst vielen Mitgliedern die zukünftige Entwicklung unseres Vereins diskutieren zu können, verbleibt...

das Direktorium

Thomas Beierlein
Egmont Woitzel
Friederich Prinz

Jahresbeitragszahlungen für 1998

Mitglieder mit ermäßigtem Beitrag: (Studenten, Rentner, Erwerbslose)	64,- DM
Ordentliche Mitglieder	96,- DM
Fördernde Mitglieder	176,- DM

auf das Konto der ***Postbank Hamburg***
PLZ: 200 100 20 Kto.: 563211-208

Inhalt

In der nächsten Ausgabe finden Sie voraussichtlich:

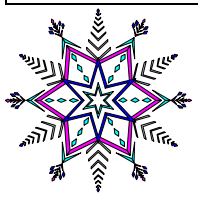
ZF spielt **WAVES**, ohne Soundkarte

Ein **HILFE**-Tool zur Arbeit mit **Fastgraf**

Die Fortsetzung von Wolf Wejgaards Aufsatz über das **HOLON**

SUCHSEL - das Werkzeug eines Lehrers, erstellt mit **WIN32FOR**

...und was Sie, liebe Leser, uns sonst noch so an Beiträgen schicken !



Ein gesundes und erfolgreiches Jahr 1998 wünscht Ihnen allen das Team der VD !

Leserbriefe

Adressen geändert...

FORTH Inc. und FIG haben ihre Adressen geändert. Die FORTH Inc. ereicht man nun so:

http://www.forth.com

Die Forth Interest Group ist jetzt zu Hause in :

100 Dolores St., Suite 183
Carmel, CA 93923
(408) 32-FORTH voice
(408) 373-2845 fax

Nach wie vor gilt für die FIG jedoch:

e-mail: office@forth.org

Forth Interest Group home page:

http://www.forth.org/fig.html

michael@malente.forth-ev.de
(Michael Kalus)

Editoriat ?

> > Editoriat?
> > Editor - ja das hatten wir
> > schon. Editor mit Beirat auch
> > schon.
> > Editoriat ist neu und klingt
> > gut :-)
>
> Wie sagt man sonst dazu,
> wenn die VD auf mehrere
> Schultern verteilt
> wird ? ;

[Expertise an]

"Editor magnificus"

Der ehrwürdige Editor und andere Fachausdrücke oder das Vokabular rund um "edit":

Lieber Friederich, dein Ausdruck "Editoriat" ist eine Wucht. Und wert durch eine Studie untermauert zu werden die ich dir hiermit zukommen lasse.

Das englische Vokabular zu "edit" lautet so:

edit
1 prepare a book, newspaper, article etc written by someone else for publication.
2 direct the production of a newspaper or magazin.
3 prepare a cinema film or tape recording by arranging parts that have previously been filmd or recorded into a suitable order.
4 arrange data for use in a computer.

editor
1 person who edits a book, newspaper, radioprogram etc.
2 person who is in charge of the preparation of a particular part of a newspaper: a news editor.

editorial
1 of an editor or editing: editorial work; an editorial decision.
2 article in a newpaper written by the

BET: **Ein Leserbrief zum Doppelcolon**

ABS: *uho@pizzicato.kbbs.org (Ulrich Hoffmann)*

Hallo lieber Fritz,

schoen, dass Du jetzt die Redaktion der VD uebernommen hast. Ich wuensche Dir viele schreibwillige Autoren. Zu einem Artikel wird es diesmal nicht reichen, aber fuer einen Leserbrief schon. In VD 3+4/97 schreibt Fred Behringer ueber das Wort Doppelcolon. Dieses Wort ist ein wirklicher Klassiker. Ich habe in meinem Archiv im fig-FORTH Goodies Package #1 folgenden Screen gefunden:

```
+-------------------------------------------------------+
|(  ::,  the immediate compilation mode        WFR-79JUN02 )  |
|: ::              ( compile and execute nameless Forth )     |
|   HERE  >R                                                  |
|   [ ' QUIT  CFA  @  ] LITERAL  ,                            |
|   !CSP  ]     ( enter compile state )                       |
|   BEGIN  INTERPRET  STATE  @                                |
|     WHILE  CR  QUERY                                        |
|    REPEAT                                                   |
|   SMUDGE  R  EXECUTE  R>  DP  !  ;                          |
|CR  ."    ::  is loaded   "  ;S                              |
|                                                             |
|The above definition '::' will accept and compile Forth      |
|words until the next ';'.    The code will then be executed  |
|once and forgotten.  This is a method to execute conditionals|
|from the terminal for testing or repetitive functions.       |
|i.e.   hex ::  5F  40  DO  I  EMIT  LOOP  ;                  |
+-------------------------------------------------------+
```

Richtig, er traegt das Datum vom 2. Juni 1979 und ist damit 18 Jahre alt! Man beachte die Aehnlichkeit zu Freds Code.

Ein Problem mag sein, dass Muell im Dictionary hinterlassen wird, wenn beim Compilieren oder waehrend der Ausfuehrung der anonymen Definition ein Fehler auftritt.

Man sollte meinen, dass genaue Systemkenntnisse noetig sind, um :: fuer ein System zu definieren, aber das stimmt nicht. Hier ist eine Definition von :: in ANS-Forth, die allerdings mit ;; beendet werden muss. Tritt beim Uebersetzen oder dem Ausfuehren ein Fehler auf, so antwortet man einfach mit: "oops", und der Muell wird entfernt.

```
: ::  S" MARKER oops  :NONAME" EVALUATE ;
: ;;   POSTPONE ;  EXECUTE  S" oops" EVALUATE ; IMMEDIATE
```

Vielleicht ist es ja auch mal wieder Zeit, ueber Forth-Systeme nachzudenken, die keinen Unterschied zwischen Interpretieren und Uebersetzen machen?

Viele Gruesse aus Kiel,
Ulrich

chief editor and often expressing her/his own opinion.

Keine weiteren Einträge sonst dazu im Englischlexikon. (Quelle: The penguine students english dictionary, 1991, London)

Und das deutsches Vokabular dazu ist:

edieren
herausgeben, veröffentlichen. (vom lateinischen "edico" ansagen, bekanntmachen, verkünden; oder "editus" verkündet, Befehle. Auch Ursprung der englischen Fassung nehme ich mal an.)

editieren
(EDV) Daten in ein Terminal eingeben, löschen, verändern. (engl. Ausdruck, eingedeutscht)

Keine weiteren Einträge sonst dazu im Duden. (Quelle: Duden Band 1, Rechtschreibung; 21. Auflage, 1996)

Du siehst, wir hatten im Deutschen bisher zwei ähnlich entlehnte Wendungen in Gebrauch. Einmal den Ausdruck "edieren" im literarischen Kontext, der dem englischen "edit" in Punkt 1-3 entspricht, und daneben in der Welt der Techniker "editieren", welches dem "edit" in Punkt 4 entspricht.
Dagegen (noch) nicht 'offiziell' eingebürgert sind "Editor" oder "Editorial". Die VD ist aber dabei, dies voranzutreiben - wir sollten sie der Dudenredaktion zu lesen geben! :-) Ganz eigen ist nun "Editoriat".
Aber Wendungen wie "Dezernent" und "Dezernat" oder "Rektor" und "Rektorat" legen solch eine Neuschöpfung durchaus nahe. Dabei bezeichnet die Endung auf "-at" zunächst den Raum in dem so ein Mensch anzutreffen ist. Und im übertragenen Sinne bezeichnen wir dann auch eine ganze Abteilung oder ein Gebäude samt seiner Mannschaft so.
Sind hingegen ausdrücklich mehrere Personen der gleichen Funktion gemeint, verwenden wir gewöhnlich die Mehrzahl der Funktionsbezeichnung. Also "Dezernenten" oder "Rektoren" um im Beispiel zu bleiben; so gesehen wäre "Editoren" der passende Ausdruck.

Die deutschen Ausdrücke im Sinne von "edit" Punkt 2 sind "herausgeben" und "Herausgeber", für die übrigen Bedeutungen haben wir keine rechten eigenen Worte entwickelt, denke ich, weil diese Techniken aus dem Ausland zu uns kamen. Eben aus England bzw. dann USA und Frankreich vornehmlich.

Bisher gebräuchlicher im Deutschen ist der "Redakteur", aus dem französischen entlehnt um das zu bezeichnen, was die Engländer "Editor" nennen. Jedenfalls übersetzt mein Langenscheidts Taschenwörterbuch von 1959 so. Die Räume und die Mannschaft sind die "Redaktion" und die Tätigkeit ist das "redigieren".
Diese Worte wurden hierzulande im 19. Jahrhundert bezeugt. Die Franzosen bildeten sie aus dem Lateinischen "re" und "agere" (zurück und treiben, handeln) und meinten damit "etwas zurücktreiben, zurückführen, in Ordnung bringen." Und das trifft eigentlich recht gut die Arbeit des "Editors", denn Autorenkollektive neigen dazu Sachverhalte zu breit und unübersichtlich darzustellen und sich nicht einigen zu können, wer was wohin schreiben darf. :-)

Die Franzosen hatten also mehr die kräftezehrende Tätigkeit selbst im Blick bei ihrer Wortschöpfung, die Engländer mehr die dazu nötige Machtausübung. Die französische Erfahrung spiegelt sich auch im Verschleiß an Editoren den die VD schon erlebt hat. Gemessen daran sollten wir vielleicht lieber von "Redakteuren" reden, das würde es uns leichter machen das Kommen und Gehen solcher Menschen zu verstehen. Irgendwie klingt "Redakteur" schon vorläufiger, nicht wahr?
Andererseits hatten wir bisher vielleicht immer die Hoffnung, eine möglichst lange stabile VD-Redaktion zu haben, einen mächtigen und unermüdlichen Menschen also, und da ist der Ausdruck "Editor" auch richtig, finde ich - ähnlich mächtig im Verein ist nur noch das kassenprüfende Forthbüro. ;-)

Somit komme ich zu Schluß meiner Ausführung: Wenn wir den "Editor" nun heimisch machen wollen, finde ich es durchaus richtig, ihm auch einen eigenen Raum zuzuweisen - das "Editoriat". Und im übertragenen Sinne steht dieser Ausdruck dann auch für die darin tätige gesamte Mannschaft.
Also meinen Segen hast du jedenfalls hiermit schon mal für deine Wortneuschöpfung. :-)

[Expertise aus]

Herzliche Grüße aus Malente,

michael@malente.forth-ev.de
(Michael Kalus)

Das ist vielleicht der längste Leserbrief, der je Eingang in die VD fand. Aber Michael Kalus' Expertise darf man Niemandem vorenthalten ;-) (fep)

Wer hat FORTH erfunden?

"Welche Frage," werdet ihr sagen, "Charles Moore natürlich!" Richtig, so wird es immer berichtet. Doch stimmt das wirklich? Oder hat uns C. M. nicht die ganze Wahrheit gesagt? War er wirklich der Erste?

James Cameron von DEC hat die ältere Literatur gesichtet - und fand tatsächlich schon 2000 vor Moore auf erste Erwähnungen von FORTH Worten. Also gab es doch direkte Vordenker. C.M. war nicht der Erste und wird sicher auch nicht der letzte sein, der FOTRH benutzte. Aber lesen sie selbst...

"... and God sent him FORTH", Genesis 3:23, KJV.
(Und Gott sandte ihnen FORTH...)

"``With the help of the LORD I have brought FORTH [...]"", Genesis 4:1, NIV.
(Mit der Hilfe des Herrn brachte ich FORTH voran... :-)

And as for software defects,
(Und was Softwarefehler angeht,)

"... all the springs of the great deep burst FORTH ...", 7:11.
(...aus allen Fluten der großen Tiefe bricht FORTH hervor...)

"... with the best the sun brings FORTH ...", Deuteronomy 33:14,
(...mit das Beste was SUN hervorbrachte ist FORTH ...)

And for a FIG meeting (BBQ) for novices,

"Ask the former generations and find out what their fathers learned, for we were born only yesterday and know nothing, and our days on earth are but a shadow. Will they not instruct you and tell you? Will they not bring FORTH words from their understanding?", Job 8:8-10, NIV.

(Frage die vorige Generation und finde heraus was ihre Väter lernten, denn wir sind alle erst gestern geboren worden als Unwissende, und unsere Tage auf Erden sind nur ein Schatten. Werden sie dir nicht die Instruktionen erklären? Werden sie nicht ihr Verständnis der FORTH Worte bringen?)

And the fight between comp. lang. forth and comp.lang.c,
(Und der Kampf zwischen c.l.f und c. l.c,)

"Day after day they pour FORTH speech; night after night they display knowledge. There is no speech or language where their voice is not heard.", Job 19:3, NIV.

(Tag um Tag erzeugen sie FORTH Sprachen; Nacht um Nacht zeigen sie ihr Wissen am Display. Es gibt keine Anwendung oder Sprache in der ihre Stimme nicht mitklingt.)

And if FORTH Inc was ever broken in to,
(Und falls sich FORTH Inc. jemals spalten sollte,)

"... in the night he steals FORTH like a thief.", Job 24:14, NIV.
(... des Nachts stahl er FORTH wie ein Dieb.)

;-)

Aus dem Englischen, von James Cameron (cameron@stl.dec.com)
Digital Equipment Corporation (Australia)

Übertragen ins Deutsche von Michael Kalus.

michael@malente.forth-ev.de
(Michael Kalus)

EMP: /DE/COMP/LANG/FORTH
BET: Win32Forth Dokumentationen
ABS: joerg.staben@cww.de

Dokumentation für Win32Forth

Tom Zimmer selbst geht auf die Frage nach Dokumentation für Win-32Forth und die WindowsProgrammierung mit Win32Forth ein.

Tom Zimmer sagt ganz deutlich, daß das Entwickeln von Anwendungen für Windows schwierig ist. Microsoft versucht zwar, mit Programmen wie Visual Basic die WindowsProgrammierung zu vereinfachen, aber dennoch bleibt ein enormes Lernpensum zu bewältigen. So als Beispiel: Win32Forth lädt ein Datei namens WINCON.DLL, die z.Zt. neuntausendvierhundert (9400) Konstanten enthält, die Sie, der Anwender, gezielt einsetzen dürfen.

Dies bedeutet auch: Wer dies nicht leisten möchte oder kann, muß sich nach einem anderen Entwicklungswerkzeug umsehen. Denn auch Win32Forth kann es nicht einfacher. Sie müssen viel über Windows lernen und zusätzlich viel über Win32Forth. Aber das wäre mit Visual BASIC oder Visual C++ genauso...

Zudem kommt für deutschsprachige Nutzer noch als Ausschlußkriterium die
englische Sprache hinzu - wobei auch dies wieder für die gesamte WINDOWSProgrammierung gilt.

Sie müssen also können oder lernen:

* Programmieren allgemein
* WINDOWSProgrammierung für Win95 und/oder WinNT
* Forth-Programmierung allgemein
* Win32Forth-Programmierung
* Möchten Sie unter Win32Forth den direkten Zugriff auf den Pro zessor nutzen, kommen noch Kenntnisse der Assemblerprogram mierung der 486er- bzw. Pentium-Prozessoren hinzu.

Um das Programmieren allgemein zu erlernen, gibt es reichlich Literatur; ebenso steht zur Programmierung von Win95 und/oder WinNT einiges an Literatur zur Verfügung - ich denke da nur an die Bücher von SAMs. An der Stelle: Zugriff auf die Microsoft Developers Network Informationen sollte man schon haben. Auch die Forth-Programmierung läßt sich literaturgestützt erlernen und das W32F selbst verfügt natürlich über einiges an Dokumentation an verschiedenen Stellen, auf die ich hier eingehen möchte.

READ.ME - Hier wird das System selbst und die Installation kurz beschrieben, auf Fehler und die grundlegenden Werkzeugen eingegangen.
WIN32FOR.NEW - geht auf die Beispielprogramme ein, als da sind:

FILEDUMP.F A File Dump program contributed by Jih-tung Pai
HELLO.F Windows interfacing without objects by Andrew McKewan
ROMCALC.F A Roman numeral calculator by Lars Krueger
WINBROWS.F File handling and window scrolling
WINDEMO.F Graphics operations in a window
WINDILOG.F User designed dialog box
WINHELLO.F Windows interfacing with objects
WINMULTI.F A multi window/ menu program by Wolfgang Engler
WINPAR.F Simple parallel port output
WINSER.F Simple serial communications program
WINVIEW.F Complete multi-file macro editor with hypertext

Der Aufruf der ANS-Definition eines Wortes wird gezeigt zusammen mit einer Beschreibung der Entwicklungsreihe des W32F, in der sich auch schon wieder Beispiele für jeweils angesprochene Themen finden.

TXT-Dateien enthalten gerne Dokumentation, sie zu lesen ist erste Bürger- Nein! Programmierer-Pflicht. So macht die über 600KB große Datei DPANS94 mit ANS-Forth bekannt und DIALOGRC zeigt die Abgründe eines Ressourcen-Compilers.

UTILDOC beschreibt die Dienstwörter des W32F, über die man einen sehr schönen ersten Zugang zum W32F bekommen kann.

Help menu ermöglicht nach dem Aufruf von Win32Forth den bequemen Zugriff auf Dokumentation in den HELP-Dateien.

F-Dateien als Quelltext-Dateien enthalten ebenfalls gerne Dokumentation; auf sie stößt man beim Durchar-

Fortsetzung auf Seite 10

Henry Vinerts berichtet über die Treffen der FIG Silicon Valley

Oktober 8, 1997

Hallo! Es scheint, daß es nach wie vor eine Reihe Leser der Vierten Dimension gibt, die an Berichten über Forth Aktivitäten in Nord Carolina interessiert sind, auch wenn diese den langsamen Weg (1200 Baud) über den großen Teich kommen. So werde ich versuchen, Euch ab und zu ein paar Zeilen zu senden. Als erstes möchte ich, bezugnehmend auf den Editorwechsel der VD, Claus zu dem gut gemachten Job gratulieren und Friederich meine besten Wünsche senden. Es scheint mir lange her zu sein (es war wohl am 29. Juni 1995), daß ich meinen ersten Brief an Herrn Prinz schrieb. Seitdem habe ich all dieEmail-Kontakte genossen, in die er mich eingeführt hat - mit all den guten Leuten in Europa. Ich muß gestehen, daß ich das August-Treffen der Silicon Valley ForthInterest Group verpaßt habe, da es in diesem August einen zusätzlichen Samstag gab (und ich den "Forth"-Samstag nicht richtig gezählt habe). Wenn ich nicht begonnen hätte am Volvo meiner Frau zu arbeiten, hätte ich mich vielleicht daran erinnert. So fragte ich einige Leute auf dem September-Treffen was im August los war, aber sie konnten sich nicht erinnern! Ich kann nur noch aufzählen, was der Plan vorsah: - einen Vortrag von John Carpenter über einen Interpreter, der zwischen konventionellem Forth und einem Fuzzy Interpreter umschaltbar ist und kontinuierlich Daten aus der Umgebung verarbeitet, - einen Vortrag von Dr. Ting über einen kleinen Forth Interpreter, der den Power-on-self-Test im Apple Macintosh Power PC durchführt und - eine Diskussion über die bevorstehende Embedded Systems Konferenz (Zu dieser Zeit hoffte die FIG immer noch einen Stand auf der Konferenz zu bekommen.) Am September-Treffen nahmen wie üblich über 20 Leute teil. Skip Carter füllte den Morgen mit einer sehr interessanten Vorführung des U.S. Robotics Palm Pilot. Ich denke, daß 3COM und USR ihn für die Werbung hätten bezahlen sollen. Er tat es aber nur für uns, um uns zu zeigen, daß "Forth zu diesem Rechner gehört" und das Neil Bridges schon ein Forth dafür geschrieben hat! Ihr könnt es von http://www.interlog.com/~~Bridges/P4TH.html holen und austesten. Der 400 Dollar kostende Pilot ist leistungsfähiger als eine alte PDP11. Für zusätzliche 100 Dollar kann man noch ein 56 kBaud Modem anstecken. Er ist mit 700.000 Exemplaren in der Öffentlichkeit schon recht weit verbreitet. Bob Smith machte die Ankündigung, daß Tom Zimmer's Version 3.5 von Win32Forth die letzte freie Version ist. Sie braucht nach wie vor eine Dokumentation und es scheint Niemanden zu geben, der diese ohne Bezahlung schreiben will. Am Nachmittag berichtete Dr. Ting über seine Erfahrungen mit dem Schreiben von CD's. Er hatte einen HIVAL JVC 2x CD-Writer mit der Toast Software benutzt, um ein paar CD's mit aufgezeichneter Musik zu beschreiben. 37 Minuten dauerte das Schreiben von 74 Minuten Musik. Der CD-Writer kostet etwa 350 Dollar, die CD's um die 3 Dollar das Stück, aber 30 % brachten Fehler beim Beschreiben der letzten Spur. Da gibt es also noch eine Menge zu lernen. Es wurde bekannt gegeben, daß es der FIG nicht gelungen ist, einen Stand auf der kommenden Embedded System Konferenz in San Jose zu erhalten. Es war wohl nicht genug Platz für die 'kleinen Fische' in diesem Jahr. Ich nahm mir einen halben Urlaubstag und ging am 30. September trotzdem zur Konferenz. Wow! Viel dichter bevölkert und viweniger verstdädlich (für mich) als das letzte Jahr. Forth, Inc. war da, obwohl sie etwas unter einem großen SwiftX-Zeichen versteckt waren. Ich war froh, daß ich sie gefunden hatte und ein Exemplar eines mit Forth hergestellten Spalding Golfballs erwischte. Schaut es Euch an unter http://www.forth.com. Drei oder vier Firmen aus Deutschland waren dort. Insgesamt waren es 227 Aussteller. Ich habe gehört, daß sie daher die kleineren Aussteller in die Flure verbannt hatten. Zu meiner Freude gab es auch zwei DOS-Händler. Einer von ihnen half mir, daß Windows CE Logo auf meinem Namensschild zu entfernen. Was mir am meisten gefiel war John Dvorak's Ansprache und seine Voraussage, daß Microsoft bis 2004 Pleite geht. So, nachdem ich nun enthüllt habe, auf welcher Seite ich stehe, wieviele von Euch wollen immer noch Bereichte von mir über den 'Großen Teich' gesandt haben?

Tschüß Euch allen!

Henry

Grüß Euch!

Meine VD 3+4/1997 kam einige Tage vor dem SVFI Oktober-Treffen bei mir an. Ich nahm sie mit zum Treffen und Chuck's Bild auf dem Umschlag zog einige Leser an. Dr. Haydon tadelte mich dafür, daß ich meine Berichte an Euch nicht in Deutsch geschrieben habe. Ich wünschte, ich könnte es. Wir hatten die übliche Anzahl Teilnehmer - zwischen 20 und 30 über den Tag verteilt; einschließlich der Studenten vom College welches uns das Klassenzimmer für das Treffen bereitstellte. Man hatte in einer Notiz alle interessierten Studenten eingeladen, zum Treffen zu kommen und eine kostenlose Kopie vom Win32Forth zusammen mit einer kurzen Einführung in Forth zu erhalten. Ich glaube, daß die beiden anwesenden Studenten etwa 1% der eingeschriebenen Studenten repräsentierten. Der eine davon hatte sich schon vorher mit Forth befaßt. Nun gut, wir brauchten länger als erwartet um das Win32Forth auf 10 bis 12 Maschinen zu installieren und wir fanden nicht einmal genug Zeit um selbst damit zu 'spielen'. Jawohl meine Damen und Herren dies ist die LETZTE freie Version von Win32Forth! Es wurde zwar bekannt, daß Forth, Inc. in Kürze PolyForth für Windows herausbringt, aber das wird Geld kosten.

Am Ende der Morgen-Sitzung quälte mich die Frage: "Verbringen wir mehr Zeit damit Werkzeuge zu bauen als diese Werkzeuge produktiv zu nutzen?" Win32Forth braucht dringend mehr Dokumentation (Und ich habe nicht einmal einen Computer auf dem es läuft. Ich verdiene meinen Lebensunterhalt nicht mit der Entwicklung von Werkzeugen, außer den AutoLISP Routinen, die meine Tagesarbeit vereinfachen). Es wird wohl eine Weile dauern, ehe sich meine 'Küche mit dem großen Toaster füllt'. Falls Ihr den Witz noch nicht gehört habt, hier ist er:

"Falls Microsoft Toaster herstellen würde, dann würde ***Toaster95*** 15.000 Pfund wiegen und 95% des Platzes in der Küche einnehmen. Er würde den Anspruch erheben der

erste Toaster zu sein, mit dem man auswählen kann wie hell oder dunkel man den Toast haben will. Jedermann würde ihn hassen aber trotzdem kaufen, da das meiste gute Brot nur noch mit diesem Toaster zu gebrauchen ist."
Skip Inskeep hat eine ganze Menge an Grafik mit dem Win32Forth erarbeitet. Er hielt darüber einen netten Vortrag und verteilte Disketten mit seinen Routinen.

Am Nachmittag war immer noch einer der Studenten da. Dr. Ting gab einen Bericht über Froth, ein Forth von Oliver Singla aus Frankreich. Es ist intern von Forth grundverschieden; assembliert statt compiliert. Colon Definitionen werden direkt in Assemblercode umgesetzt. Es läuft unter Win95 als 32-Bit System, besitzt einen eigenen Editor "FrostED" und eine Anzahl weiterer 'Goodies'. Ich weiß das "Froth" im Deutschen "Schaum" bedeutet, aber in meinem Land macht 'frothing" keinen guten Eindruck. Ich hoffe, daß Mr. Singla's Produkt kein Opfer seines Namens wird, wie wohl es 'Fifth' gegangen ist (s. S.38 VD 3+4/97).

Das war es für diesmal, Leute. Ich habe genug geschwätzt und vertraue darauf, daß ihr nur das druckt, was akzeptabel ist.
Tschuess,
Henry

Die Übersetzung der Berichte wurde von Thomas Beierlein angefertigt.

Hinweis: Auf Nachfrage hat Tom Zimmer die Angaben bestätigt, denen zufolge es keine Version 3.6 von WIN32FORTH mehr geben wird (kann). Alle zukünftige Arbeiten an WIN32FORTH werden 'Bugfixes' sein. Die Arbeit an den Bugfixes konzentriert sich im wesentlichen auf WINVIEW.

(fep)

Fortsetzung von Seite 8

beiten der Quelltexte, die dem W32F beiliegen.

486ASM.DOC beschreibt die Leitungsmerkmale des 486/586er Assemblers, seine aktuellen Fehler sowie seine Installation.

Dann fanden sich im Internet noch die WIN32FOR .FAQ Dort wird im Frage/ Antwort-Verfahren auf die verschiedensten Themen eingegangen:

heap-Größe und Microsoft's Window API constants (from wincon.dll)
Werkzeuge zur Fehlersuche; source level debugging information
Interpretieren der Eingabezeile bezgl. des IEEE floating point number formats
Groß-/Kleinschreibung in W32F und beim Zugriff auf die WinAPI
Zeiger auf den Forth-Stack und callbacks
Löschen von Stack-Werten
Speichermodell des W32F; Speicherzugriffe von W32F aus
Erzeugen von Programm- (EXE)-dateien
Fehlender Zugriff auf VBX's
Bislang fehlende OLE-Unterstützung, das sowieso durch Microsoft ActiveX ersetzt werden wird.
Umgang mit den Prozessor-Registern
Gestaltung von Fenstern
Erzeugen von neuen Dialog-Boxen ohne Visual C++.
Skalierung von VGA/SVGA-Text
Umsteigen von F-PC auf Win32Forth Literatur für Win32Forth
Literaturempfehlung für Win95 Programmierung

SAMS makes a book called Win32 API (ISBN 0-672-30364-7 or LCCN 93-84382)

NOTE: June 4th, 1997 Addison Wesley Longman Inc publishes a book called "Win32 Programming" by Brent E. Rector and Joseph M. Newcomer with an ISBN number of 0-201-63492-9. Published December 1996

Unterschied der Objektorientierten Programmierung von der konventionellen Programmierung - Dokumentation zur Objektorientierten Programmierung; Verweis auf das MOPS Handbuch das MacIntosh - Umgang mit - und Fehlersuche in - den Methoden der Objektorientierten Programmierung
Zugriff auf die API32.HLP Dateien, die im kommerziellen Visual C++ enthalten sind.

ALSO! ES IST ALLES DA.

VIEL SPASS!!

Jörg Staben

Fehlende, unzureichende, schwer zugängliche Dokumentationen und Hilfen zu Win32For werden immer wieder als DIE Kriterien genannt, die dem Anfänger und/oder Quereinsteiger den Start mit Tom's Forth unnötig erschweren. Jörg's Zusammenfassung gibt in diesem Sinne nicht nur dem Martin Bitter (siehe Beitrag "Im Gleichschritt") wertvolle Hinweise, wo welche Hilfen 'greifbar' sind. Vielen Dank, Jörg.

(fep)

Fehlersuche in Win32Forth

Win32Forth verfügt mit dem Kommano DEBUG über eine ähnliche Unterstützung der Fehlersuche wie das F-PC. Die Taste '?' (Fragezeichen) zeigt während der FS eine Liste der verfügbaren Kommandos an.

Um einen Fehler in einem Wort zu suchen, wird DEBUG folgendermaßen eingesetzt:

DEBUG WORDS

Der Debugger schreibt einen breakpoint in die erste Zelle der Definition WORDS, und wartet darauf, daß WORDS ausgeführt wird. Da aber WORDS bis jetzt nicht aufgerufen wurde, passiert trotz des aufgerufenen Debuggers erstmal nichts.

NOTE: Sobald Sie mit der FS fortfahren, zeigt der systemeigene Editor WINVIEW gleichzeitig den Quelltext der Definition WORDS an. Mit Ctrl-PgUp kehren Sie von dort zurück.

Damit der Debugger startet, muß das Wort mit dem bereits eingetragenen Breakpoint aufgerufen werden: WORDS !
In diesem Fall wird die Definition zusammen mit einem Parameter (!) aufgerufen; der Debugger wird akti-

Fortsetzung auf Seite 13

Einladung zur

Mitgliederversammlung

der Deutschen Forthgesellschaft e.V.

Am **26. April 1998, 9:00 Uhr**
Hotel Dampfmühle
Krefelder Straße 9
47506 Neukirchen-Vluyn

Tagesordnung:

1. Begrüßung der anwesenden Mitglieder
2. Wahl des Schriftführers
3. Wahl des Versammlungsleiters
4. Ergänzungen zur Tagesordnung
5. Bericht des Direktoriums
   - 5.1 Rund um die VD (F. Prinz)
   - 5.2 Zum Forthbüro (E. Woitzel)
   - 5.3 Mitgliederentwicklung und Kassenstand (U. Schnitter)
   - 5.4 Vorhaben des Direktoriums (T. Beierlein)
6. Entlastung des Direktoriums
7. Senkung der Mitgliederbeiträge
8. Verschiedenes

Entsprechend unserer Satzung können weitere Tagesordnungspunkte auf Antrag einzelner Mitglieder von der Mitgliederversammlung durch Abstimmung auf die Tagesordnung gesetzt werden.

für das Direktorium

Friederich Prinz

Im Gleichschritt ...

Cursorposition in der Forthkonsole bei unterschiedlichen Auflösungen

von Martin Bitter
Möllenkampweg 1a, 46499 Hamminkeln,

Bei hohen Auflösungen wird im Konsolenfenster von Win32For das Caret nicht richtig positioniert. Eine Möglichkeit das zu ändern wird hier gezeigt.

Stichworte: Win32For, Konsole, Cursor; Auflösung >1024x768

Ich betreibe Win32For unter WWF311 mit der Erweiterung Win32S. Meine Standardeinstellung für den Bildschirm ist 1024 x 748.
Bei diesen Voraussetzungen kommt es zu folgenden unschönen Erscheinungen:

1. Der Standardzeichensatz ist extrem klein.
2. Der Cursor (die Texteinfügemarke) hat eine andere Schrittweite als der verwendete Zeichensatz.
3. Bei Fehlermeldungen wie **Accessviolation** sind fett ausgegebene Zeilen nicht lesbar, da Zeichen- und Zeilenhöhe nicht überein stimmen (halbhohe Zeilen nach >bold).
3. Die von GetColRow zurückgegebenen Parameter stimmen nicht mit den tatsächlichen Gegebenheiten überein. Die Parameter, die an SetColRow übergeben werden, führen zu (von mir) unerwarteten Größen des Konsolenfensters.

Probieren Sie es aus: Rufen Sie in einer hohen Auflösung einige Male hintereinander GetColRow SetColRow auf.
Tippen Sie irgendeine Zeile ein, und versuchen Sie den Cursor unter einen Buchstaben nahe am Zeilenanfang zu setzen.
Provozieren Sie einen ACCESVIOLATION-Fehler.

Lesenswert (12x20 Pixel)

Wird der aktuelle Zeichensatz verändert, was bei mir wegen der Lesbarkeit unbedingt notwendig ist, so "weiß" die Konsole nichts über die veränderte Zeichensatzgröße, diese muss ihr mit dem Wort setcharwh mittgeteilt werden.
Eine Möglichkeit der Verbesserung? besteht in den unten gezeigten Codezeilen. Dort werden die Größen des Zeichensatzes per Hand über setcharwh an charwh weitergereicht. (ein call gettextmetrics etc. scheidet leider aus. vgl. unten)

Die so veränderte Zeilenlänge "bemerkt" die Konsole verständlicherweise? ebenfalls nicht. Es kann daher vorkommen, dass ein CR weit vor dem rechten Fensterrand durchgeführt wird, bzw. dass der Cursor ohne CR unter dem rechten Scrollbalken verschwindet.
Die Lösung besteht darin, vor dem Ändern eines Zeichensatzes die Fenstergrösse in Pixeln zu errechnen, sie zu merken, den Zeichensatz zu ändern, und abhängig von den gemerkten Pixelmaßen eine neue Konsolengröße in Cols und Rows zu setzen.
Diese Aufgabe erfüllt das Wort CHANGE_FONT, dem die ID des neuen Fonts und seine Maße übergeben werden.
Versuche, die Maße des gesetzten Fonts Mithilfe des API-Aufrufs von GetTextMetrics zu erfahren schlugen fehl. Diese Funktion liefert meist richtige Werte, aber bei ANSI_FIXED_FONT werden 9x12 Pixel gemeldet, statt 8x12 Pixel, wie es richtig wäre. Andererseits hat sie mir gute Näherungen geliefert, musste ich doch die Fontgrößen empirisch ermitteln.
Die Sprünge beim Ändern der Fontgröße gehen auf das Aufrufpaar GetColRow-SetColRow zurück. Sie sind geringer als beim Original, trotzdem versuche ich sie durch verschiedene Korrekturwerte zu vermindern.
Ein weiterer Nachteil ist, dass der Zeichensatz für das ganze Konsolenfenster geändert wird. Verschiedene Zeichensätze gleichzeitig - das geht also nicht mehr.
(Das passiert im Original übrigens auch, wenn man über die Grenzen des sichtbaren Bereichs hinaus scrollt.)
Ein kleine Dreingabe ist ein neuer Zeichensatz "Schmal" der durch >small aufgerufen wird.

Ursachenvermutungen

(Hoffentlich habe den Fritz Prinz richtig verstanden.)
Tom Zimmer hat einen Lader Win32For.exe mit Visual C++ Version 2.0 - 2.2 geschrieben. Der eigentliche Programmcode liegt in der gleichnamigen Win32For.img (Imagedatei = reines Forth). Der Programmcode der z.B. mit TURNKEY erzeugt wird, liegt in einem File NAME.IMG, die Datei NAME.EXE ist, vereinfachend gesagt, eine Starterdatei, die einige Verwaltungsarbeit erledigt und dann in den

Code der Imagedatei springt.
Unter anderem liegt in der EXE-Datei eine Tabelle mit grundlegenden (primitiven?) API-Aufrufen. Diese werden über die FORTH-Funktion XCALL als externe Funktionen angesprungen.
Die Maße der Forthkonsole werden über solche xcall's verwaltet.
Die Funktion GetColRow ruft die API-Funktion getcolrow_x (xcall 23) auf, diese liefert abhängig vom aktuellen Font die Größe des gesamten Fensters ohne evtl. Scrollbalken in Buchstabenhöhe und -breite zurück. Dies geschieht mit einem Rundungsfehler, wenn die Buchstabenbreite nicht glatt in der Breite des Randes und Scrollbalkens enthalten ist.
Das Wort SetColRow ruft die Funktion RESIZE (xcall 32) auf, die ihrerseits pixelorientiert ist. D.h. die Daten Col bzw. Row müssen in Pixelmaße umgerechnet werden. Dabei geht T. Zimmer von einer festen Buchstabengröße von 8x13 Pixeln aus, bildet daraus die Größe des zugehörigen Fensters und übergibt sie an RESIZE, welches nun die Breite des Scrollbalkens und des Randes zu dieser Größe addiert und einen Redraw des Fensters auslöst.
Das führt dazu, daß bei wechselndem Aufruf von getcolrow-setcolrow die Fenstermaße in großen Sprüngen verkleinert werden.

Ohne Tritt ....

Je länger ich versuche, Windows mit Win32For zu erlernen, desto dümmer komme ich mir vor. Die große Vielfalt, die es mir bietet, macht es mir sehr schwer, DIE richtige Lösung zu finden und bei Vielem bleibt das dumpfe Gefühl, die richtige, einfache und elegante Lösung nicht erkannt zu haben.
Bei diesem Code springen z.B. die Fenstermaße wenn man zwischen >norm, >small und >bold wechselt, bleiben aber bei zyklisch wiederholtem Aufruf konstant..

Fortsetzung von Seite 10

```
28 Value x-korrekt
0  Value y-korrekt
: setcolrow ( cols rows -- )        \ ein neues setcolrow
    charwh                          \ aktuelle Zeichensatzdaten
    rot * y-korrekt + -rot          \ Z-Höhe mal Zeilen plus Korrektur
    * x-korrekt + swap              \ Z-Breite mal Sp. plus Korrektur
    32 xcall drop ;                 \ externe Funktion RESIZE rufen

: colrow>pix ( -- h w )                         \ Konsolenmaße in Pixeln
    getcolrow charwh rot * y-korrekt +  \ Höhe
    -rot * x-korrekt + swap ;                   \ Breite; passende Reihenfolge

: change_font ( id w h -- )             \ neuen Font auswählen
    colrow>pix >R >R                    \ Fenstermaße in Pixeln merken
    rot set-font setcharwh              \ Font setzen, Z-größe aktualisieren
    R> R> 32 xcall drop ;               \ neue Col und Row setzen

: my_bold ( -- )                        \ ein Zeichensatz FETT
    OEM_FIXED_FONT 10 20 change_font ;

: my_norm ( -- )                        \ lesbarer Zeichensatz NORMAL
    DEVICE_DEFAULT_FONT 12 20 change_font ;

: my_small ( -- )                       \ und noch ein neuer: SCHMAL
    ANSI_FIXED_FONT 8 12 change_font ;

defer >small
' my_bold is >bold
' my_norm is >norm
' my_small is >small

>norm 60 30 setcolrow
```

viert und zeigt den Stackinhalt vor dem Eintritt in die Definition an. Zugleich wird der Quelltext der aufgerufenen Definition im systemeigenen Editor WINVIEW angezeigt.
Wenn Sie die Stack-Anzeige (zuerst wohl das Wort 'empty') nachverfolgen, sehen Sie 'const 0', das CONSTANT mit dem Wert 0 bedeutet.
Drücken Sie nun im Konsolen-Fenster <cr>, so sehen Sie '[1] 0' als Anzeige für einen Stackwert, die Null. Der Debugger zeigt dann 'TO WORDS-CNT' an. Dies ist die Stelle, in der die Null gespeichert wird, nämlich der Value WORDS-CNT.

Jetzt sollten Sie die '?'-Taste drücken, um sich mit den vorhandenen Kommandos im Debugger vertraut zu machen. Besonders die Kommandos 'N' (nest) und 'U' (unnest) sind sinnvoll, um tiefer in eine Definition einzusteigen und von dort zurückzukehren. So können Sie den Punkt im Programm gezielt erreihen, der Sie bei der Fehlersuche besonders interessiert.

Das andere Wort zur Fehlersuche DBG arbeitet ähnlich wie DEBUG, ruft aber das Nachfolgende Wort sofort auf. Dafür müssen Sie natürlich den entsprechenden Rahmen schaffen, also zumindest die erforderlich Anzahl Parameter

Fortsetzung auf Seite 18

Win32Forth: Programm mit mehrsprachiger Oberfläche

Wie gestaltet man im Forth-Programm die Ausgabe der Steuerungstexte von Windows?

Von Ulrich Richter
Oberwallstraße 4 ; D 47441 Moers

Diese Arbeit entstand zusammen mit einer Arbeitsprobe im neuen Win32Forth. Programmiert wurde ein Schiebespiel. Die Übersetzung soll Kindern aus der Familie des Autors helfen, sich an den Umgang mit dem Computer zu gewöhnen. Sie wohnen in einem fernen Land und lernen das Deutsche erst später oder gar nicht. --
Das vollständige Programm befindet sich auf:
http://www.taygeta.com/pub/forth/Win32For

Will man unter Windows Ausgaben machen, so sind die zur Bedienung des Systems gehörigen Texte mit den Befehlen des API als Stringkonstanten zu übergeben. -- Eine Auswahlmöglichkeit unter vorhandenen Texten in verschiedenen Sprachen ist nicht vorgesehen. -- Das ist auch nicht nötig, denn Windows gestattet außerdem, daß die an der Oberfläche bereits vorhandenen Texte nachträglich verändert werden können. Dafür hat das API die Befehle:

SetWindowText, SetMenuText, SetDlgItemText

Es gibt bei Windows allerdings auch noch andere Texte, 'die zum System gehören' (zB das Wort 'Schließen'). -- Diese sind für den Programmierer überhaupt nicht erreichbar. Sie erscheinen immer in der Sprache des jeweiligen Windows-Systems. Bei diesen kann der Übersetzer nur hoffen, daß das Ausführungssystem bereits in der Sprache installiert ist, auf die das Forth-Programm gerade eingestellt werden soll.

Der folgende Artikel stellt Möglichkeiten dar, wie man alle vom Programm benötigten Texte in andere Sprachen überträgt, so daß sie beim Windows abgeliefert werden können. Außerdem wird gezeigt, wie man die Texte in die Windows hineinbringt. Das Verfahren kann aber auch für alle Texte der Programmausgabe (Bedienen der Client Area) angewendet werden.

Es wäre sicher lästig, wenn der Übersetzer das ganze Programm durcharbeiten müßte, und immer dort, wo der Programmierer einen Text abgesetzt hat, in richtigem Forth-Code seine Übersetzung einfügen müßte. Vielmehr ist es nötig, alle Texte des Programms zusammen mit ihren Übersetzungen in einem einzigen Modul zu erfassen und nach einer einfachen, für den oder die Übersetzer leicht verständlichen Vorschrift abzulegen.

Außerdem soll die eigentliche Programmierung möglichst wenig durch die Übersetzung beeinflußt werden; Auch für den Programmierer müssen einfache Forth-Worte zur Verfügung stehen, die den üblichen Worten der Stringverarbeitung genau entsprechen, und die dennoch die richtige Übersetzung (Auswahl des richtigen Textes aus der Stringtabelle für alle zur Verfügung stehenden Sprachen) herbeiführen. --

Ein Beispiel:

In der zentralen Stringtabelle werden von den Übersetzern die vom Programm zu benutzenden Strings in allen Sprachen bereitgestellt:
Will man nun den String 'Schwarz' in eine andere Sprache übersetzen, so sind zwei Veränderungen nötig:

```
Synonym  tt  ,"Text"    \ Anfang der Stringtabelle

Create  Hell
tt  "Weiß"          tt  "White"       tt  "Blanc"
Create  Dunkel
tt  "Schwarz"       tt  "Black"       tt  "Noir"
```

Anstelle des eigentlichen Strings wird ein Symbol angezogen; Dieses Symbol steht vor dem auszuführenden Forth-Wort .

Das auszuführende Forth-Wort (zB s" oder z") wird verändert, indem ein 'm' davor gesetzt wird, was hier Mehrsprachigkeit bedeuten soll.

Mit diesen beiden einfachen Regeln kann man sehr einfach mehrsprachig programmieren. --

Zwei Beispiele:

s" Schwarz" —> Dunkel ms"
z" Weiß" —> Hell mz"
*u*sw.

Weiter oben im Programm, kurz nach der entscheidenden Abfrage, steht irgendwo im Programm noch die wesentliche Zeile:

2 to Sprache

So ist es ohne Weiteres klar, daß hier die 'm-Stringbefehle' mit Übersetzung in die zweite Sprache durchzuführen sind. ----

In diesem Beispiel wäre es natürlich auch möglich, das deutsche Wort 'Schwarz' oder 'Weiß' als Symbol zu benutzen, aber man kann dieses nicht zur Regel machen, denn die zu übersetzenden Strings enthalten oft Blanks und unübliche Sonderzeichen.

Will man am Win32Forth nur behutsam ändern, dann ist es richtig, einen zusätzlichen, neuen 'VALUE SPRACHE' wie folgt zu wählen:

0 Vordefiniert; Keine spürbare Veränderung des Originalcodes.
1 Übersetzen in die erste Sprache: Hier Deutsch
2 Hier Englisch
3 Hier Französisch

SPRACHE ist also immer auf Null eingestellt; Nur im Bedarfsfall wird SPRACHE überhaupt angefaßt.

Ansatzpunkt für die nun zu erläuternde kleine Programmieraufgabe ist das bereits im Win32Forth vorhandene Wort ,"Text" aus dem Modul Primutil.f . Dieses Wort legt nämlich die anschließende String-Eingabe kontinuierlich in Strukturen der Form

Byte Anzahl
Chars Der String
NULL

ab. Die Speicherung beginnt 'here', der Zeiger wird automatisch um Anzahl 2 + erhöht. -- Damit sind also beide Grundformen der Stringverarbeitung, nämlich 'Counted String' und 'z-String' bereits fertig abgedeckt. –

Bei der Übersetzungsaufgabe wird dieses Wort gleich in zweifacher Hinsicht benötigt:

Zum Ablegen der Strings in der Übersetzungstabelle,

Zum Bereitstellen der Übersetzung an den Stellen, wo der Forth-Code Stringeingaben erwartet. –

Hierbei ist zu beachten, daß dieses Wort vom Autor des Win32Forth überall dort angewendet ist, wo Menus oder Buttons oder sonstige Beschriftungen nötig sind --- Das heißt: Die Auswahl der Stellen, wo am Code des Forth einzugreifen ist, wird dadurch erledigt, daß das Wort ,"Text" verändert wird.

Zuerst ein ganz kurzes Stück Code, das die Adresse eines Strings von der ersten Sprache auf die Adresse der Zielsprache erhöht:
Mit diesem Wort werden alle Übersetzungsaufgaben erledigt.

```
x  VALUE Sprache \ Vom Startdialog richtig gesetzt
\ ==============================

: mm   ( Addr0 -- Addrn )   \ Sprache 1 —> Zielsprache
  0 >r
  BEGIN
    r> 1+ DUP >r Sprache  =
    IF
       r> DROP EXIT       \ Fertig
    ELSE
      DUP C@ 2 + + \ Nächste Sprache
    ENDIF
  AGAIN
  ;
```

Die Änderungen am Wort ,"Text" sind genau so einfach. -- Solange SPRACHE = 0, läuft unverändert das Original Win32Forth. Wenn SPRACHE <> NULL ist, muß der Text in der richtigen Sprache 'here' aufgesetzt werden.

Im Bestreben, das Wort ,"Text" sowohl original (zum Ablegen here) als auch zur Übersetzung (Aufsetzen der richtigen Sprache aus der Stringtabelle, here) zu benutzen, sind hier mehrere ganz wesentliche Nachteile in Kauf genommen worden:

Bei SPRACHE = NULL muß auf ,"Text" ein String nachfolgen, der mit Quote abzuschließen

ist. Vor dem Wort darf kein Symbol stehen.

Bei SPRACHE <> NULL muß ein Symbol vor dem Wort ,"Text" stehen. Und es darf kein String nachfolgen.

Am Wort ,"Text" ist so viel geändert, daß das Forth neu kompiliert (MakeForth) werden muß.

Wenn man bei diesem 'SPRACHE-Komplex' Fehler macht, werden folgende Meldungen ausgegeben:

Bei SPRACHE <> NULL mit nachgeschobenem String (und ohne Symbol) erscheint 'Stack depth increased' und Win32Forth zeigt auf Modul und Zeilennummer --

Bei SPRACHE = NULL mit vorlaufendem Symbol (und ohne String) gibt es die Fehlermeldung 'Undefined Symbol ...' mit Modul und Zeilennummer.

Die Stringbefehle des Forth befinden sich in den Moduln FKernel.f und Primutil.f. Wie viele davon nun wirklich zur mehrsprachigen Bearbeitung übertragen werden müssen, hängt vom aktiven Wortschatz des Programmierers und vom Problem ab. Es kommt nur darauf an, das Wort 'mm' richtig anzuwenden. –

Zwei Beispiele:

```
: ms" ( Addr0 -- Addrn len ) \ Mehrsprachiges s"
    mm DUP 1+ SWAP C@ ;

: mz" ( Addr0 -- Addrn ) \ Mehrsprachiges z"
    mm 1+ ;
```

Somit sind 'forth-seitig' die Voraussetzungen gegeben, alle Texte zur Übergabe an Windows zu übersetzen.

Bei diesen Übersetzungen gilt es allerdings noch zu überlegen, wann diese durchgeführt werden sollen. Es gibt zwei wesentliche Möglichkeiten:

SPRACHE wird mit einem modalen Vorab-Dialog erfragt. Die Hauptfenster werden erst dann mit FLOAD geladen, wenn SPRACHE den endgültigen Betrag hat. -- (Keine Besonderheiten beim On_Init ...Übersetzung beim FLOAD...) .

```
: ,"TEXT"
   Sprache   0=
     IF
   ( -<"text">- )  \ parse out quote delimited text and compile
      \ it at here  NO EXTRA SPACES ARE NEEDED !!!

            ...   der Originalcode des Win32Forth   ....

     ELSE ( Addr0 -- )\  Adr0  =  Symbol auf  Sprache 1
       mm                \  Die richige Sprache erscheint here
       DUP  C@  2  +  here  SWAP  CMOVE
       here  C@  2  +  ALLOT
     ENDIF
   ;
```

Es werden keine besonderen Vorkehrungen getroffen, um das Laden des Hauptfensters so lange aufzuhalten, bis SPRACHE richtig gesetzt ist. Alle FLOADs werden mit SPRACHE = NULL durchgeführt. Alle nötigen Texte werden während des On_Init mit unseren Forth-Befehlen übersetzt und mit BGs Set-Irgendwas-Befehlen nachträglich in die Items gepatched (... Übersetzung beim On_Init...) .

Im Folgenden soll die Programmierung beider Varianten an Hand eines fiktiven Beispiels gezeigt werden. – Im diesem Beispiel soll ein MenuItem mit dem Text Hell (unsere Stringtabelle oben) beschriftet werden. Auszuführen ist das Wort DoHell.

Fall 1 --- Übersetzung beim FLOAD

Zunächst darf nur so viel geladen werden, daß die Sprachabfrage geht. Dann wird die Sprache abgefragt. Anschließend wird mit 'richtigem' Wert SPRACHE weiter geladen.

```
.
FLOAD   GetSprache
Start:  DieSprache
FLOAD   MainProblem
Start:  DasProblem
.
```

Einerlei wo und wie die Anweisungen in der Quelle stehen: Es kommt nur auf die Reihenfolge der FLOADs und Starts an. -- Das Frageobjekt ist definiert als:

```
:Object DieSprache <super Dialog
...
;Object
```

Damit dies so einfach geht, ist zum Objekt DieSprache vorher ein Template erzeugt worden. Solche Templates gestatten zahlreiche Möglichkeiten; -- Über die Einbindung und Übersetzung fremd (außerhalb des Win32Forth) erzeugter Templates wird am Ende dieses Aufsatzes noch eingehender berichtet.

Im Modul MainProblem stehen vor dem Objekt DasProblem die Menubeschreibungen. Dazwischen befindet sich die Zeile:

```
Hell MenuItem DoHell ; \ SPRACHE <> 0
```

Fall 2 --- Übersetzung beim On_Init

Gleiche Ausgangssituation wie oben

```
FLOAD   GetSprache
FLOAD   MainProblem
Start:  DieSprache
```

Hier wird erst gestartet, nachdem das letzte FLOAD fertig ist. -- Das Objekt DieSprache kann ein gewöhnliches Window sein und braucht kein Template. Die Sprache wird abgefragt. Beim (oder nach dem) Schließen des Objekts DieSprache wird das das Objekt DasProblem gestartet. – Vor dem Objekt DasProblem sind irgendwo die Menueinträge. -- Die Zeile für unseren Eintrag lautet hier:

```
:MenuItem ObjHell "VorabHell" DoHell ;
```

Es handelt sich jetzt um ein MenuItem mit Objektnamen. Deshalb steht ein Colon vor dem Wort MenuItem. Der Objektname (hier 'ObjHell') wird gebraucht, wenn der Text des Objekts geändert werden soll. Zur FLOAD-Zeit ist der Menutext noch 'VorabHell'.

Im On_Init von 'DasProblem' befindet sich noch eine Tabelle mit der Übersetzung aller Items. Sie enthält auch die Zeile:

```
Hell mz" SetMenuText: ObjHell
```

Zu allem Unglück ist beim Schreiben des Moduls Menu.f die Klasse :POPUP (Klasse POPUP mit explizitem Objektnamen) vergessen oder nicht für nötig gehalten worden. Will man die Texte von POPUPs übersetzen, dann braucht diese jedoch:

```
INTERNAL

:Class :POPUP <super POPUP

:M SetMenuText: ( z$ -- )
        rel>abs
        pid
        MF_BYCOMMAND MF_STRING or
        pid MenuHdl
        Call ModifyMenu ?win-error
        ;M
;Class
EXTERNAL
```

Einzelheiten siehe Modul Menu.f .

Damit ist auch der direkte Kontakt zwischen Forth und Windows klar.

Will man schließlich mit einem '.res File' fertige, größere, fremd erzeugte Resourcen einfügen, so werden deren Texte natürlich nicht übersetzt. Die einzelnen Items sind überhaupt nicht im OOP des Win32Forth enthalten. Die Übersetzungen können nur mit nachträglichem Patchen in die einzelnen Items gebracht werden. Dieses Patchen muß mit den Befehlen des API durchgeführt werden:

Alle zu übersetzenden Strings haben Namen und ID (also keine Statics mit -1).

Alle Items sind sorgfältig nach links oder rechts ausgerichtet weil sich die Länge ändern wird.

Das .res-File und das .h-File haben denselben Filenamen. Sie werden mittels load-dialog eingelesen. (Modul Dialog.f)

Der Text für die einzelnen Items wird mit den hier beschriebenen Mitteln übersetzt.

Bevor der Text mit SetWindowText in die richtigen Felder übersetzt werden kann, muß erst noch das Windows-Handle jedes einzelnen Items erfragt werden.

Im folgenden Code wird der Text von 'Hell' der obigen Stringtabelle in das fiktive DialogItem 'WinzigHell' einer komplizierten Resource übersetzt:

```
: TrlItem ( LPZStr ItemName -- )
   \ Übersetzt Items
  Call GetDlgItem
  Call SetWindowText DROP
  ;
```

Hell mz" WinzigHell TrlItem \ Das Item

Damit ist auch die Bearbeitung von Files mit Resourcen klar.

Fortsetzung von Seite 13

auf den Stack legen. Dies gilt natürlich auch für DEBUG, wo das zu untersuchende Wort auch mit Parametern versorgt werden muß. Das Beispiel sieht für DBG so aus: DBG WORDS !

Der Debugger beginnt sofort mit der Fehlersuche in WORDS; dabei wird das Zeichen '!' vom Eingabestrom an WORDS übergeben.

Eine Übersicht über die Debug-Kommandos:

ENTER/SPACE Programmausführung im Einzelschritt

ESC/Q Debugging beenden, unbug, Abbruch zurück zum Forth.
C kontinuierliche Programmausführung bis auf Tastendruck
D Debugging beenden, Program bis zum Abschluß ausführen
F Eine Forth Kommandozeile eingeben
H Umschalten zwischen Hex und Decimal Zahlen
J Nächstes Wort überspringen; sinnvoll zum Verlassen von LOOPs
N Springt in eine Colon oder Does> Definition
P Proceed to initial breakpoint location again
^P Proceed to the current program point again
R Zeigt den Return stack
U Springt zur aufrufenden Definition zurück
W Ermöglicht die Eingabe des watch-Kommands

Jörg Staben

Holon*forth*©

Von Wolf Wejgaard auf Holon's eigene Web-Seite aufmerksam gemacht, hat mich meine letzte Surf - runde nach **http://WWW.HOLONFORTH.COM** geführt. Um das vorweg zu nehmen: ***diese Seite ist einfach gelungen*** ! Wolf Wejgaard präsentiert unter dieser Adresse auf mehreren Seiten sein Holon auf eine ansprechend schlichte und gleichzeitig eindrucksvoll überzeugende Weise. Keine schreiend bunten Farben, keine marktschreierische Aufmachung, keine Blenderei - statt dessen kurze, einprägsame Informationen zu Holon und zu FORTH. Links zu internationalen Adressen sind vorhanden. Wer mag, kann über diese Links direkt Kontakt zu Chuck Moore und Elizabeth Rather aufnehmen. Und ein System zum Testen fehlt natürlich auch nicht. **HTEST86D.ZIP** läßt sich als File herunterladen. Das ist, wie Wolf mir geschrieben hat, im Prinzip die in Moers 'aufgenordete' Version von HOLON, die mit einer kleinen Einschränkung bereits der Vollversion dieses ausgezeichneten Werkzeugs entspricht.

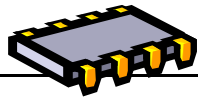
Sehen Sie am besten bei Ihrer nächsten Surfrunde einfach selbst dort hinein. Es lohnt sich !

Friederich Prinz

Informatik-Handbuch

Auf 961 Seiten enthält dieses Buch in einer sehr kompakten Form alles, was Heute zum Thema Informatik gelehrt und gelernt wird. Auf einem ausgesprochen hohen Niveau beschreibt das Handbuch Informatik in 40 Kapiteln die großen Teilgebiete : Theoretische Informatik - Daten - Technische Informatik - Praktische Informatik - Angewandte Informatik - Wirtschaftsinformatik - Normen und Spezifikationen.
Ein umfangreiches Stichwortverzeichnis ermöglicht das rasche Auffinden aller relevanten Informationen wie zum Beispiel zu Themen wie "Algorithmen und Datenstrukturen", "Verteilte Systeme" oder "Transformationen, Projektionen, Clipping".
Das auf Universitätsniveau erstellte Handbuch ist ein Wissenskompendium, das seinen festen Platz auf dem Schreibtisch der Profis der unterschiedlichsten Bereiche haben sollte.

HANSER Verlag
ISBN 3-446-18691-3
98,- DM



Zen Floating Point von C.H. Ting

Vereinfacht nach Martin Tracy.
Aus FD Januar 1997.
Für Turbo-Forth bearbeitet und erweitert

Von Fred Behringer
Planegger Straße 24, 81241 München

Jahrelang habe ich mir überlegt, ob ich mir zu meinem 286er noch einen 287er kaufen soll. Ich habe es nie getan. War mir zu teuer. Stattdessen habe ich damals Ideen von Marc Petremann und anderen aus der französischen Forth-Gruppe aufgegriffen und das Gleitkomma-Paket aus Turbo-Pascal 3.0 so verwendet, daß die einzelnen Bestandteile von Turbo-Forth aus aufrufbar wurden. Softwaremäßig simuliert, wohlgemerkt.

Heute habe ich schon seit einiger Zeit einen (einen? - mehrere) 486er. Die sind ja erschwinglich geworden und werden einem auf den Computer-Flohmärkten nachgeworfen. Da hat man nun aber einen Gleitkomma-Prozessor als Beigabe und weiß nichts damit anzufangen. Warum? Weil man meint, es sei zu schwer, ihn zu programmieren. Nichts da ! Ganz leicht ist es - wenn man nur den nötigen Anstoß bekommt.

Ich fand diesen Anstoß letzte Woche bei C.H.Ting, einem in der amerikanischen FIG und inzwischen wohl auch bei uns wohlbekannten Autor "vom Fach" (Mathematik), der sich in einem Artikel in der FD vom Januar 1997 die Mühe gemacht hat, einen Vorschlag von Martin Tracy aufzugreifen und die Gleitkomma-Befehle des 287 bis "587" für Forth aufzubereiten. Die Floating-Point-Assembler-Befehle in CODE-Definitionen gepackt, den 8 Einträge fassenden Floating-Point-Stack ausgenutzt, eine 2VARIABLE zum Übertragen eingeführt, einen FBUFFER zum FDUMPen eingerichtet, ein paar E/A-Befehle hinzugefügt (F@ F! F.) - und schon hat sich's.

(C.H.Ting ist vielen von uns noch aus Dr. Dobb's Zeiten bekannt. In F83, ZF, F-PC und auch in Turbo-Forth sind seine Beiträge eingearbeitet und weitergereicht worden.)

Ich habe das Programm von C.H.Ting abgetippt und ausprobiert. (Im folgenden spreche ich immer von Turbo-Forth (16-bittig). Die eben genannten anderen Forths gehen aber sicher auch.) Natürlich ging auf Anhieb gar nichts. Aber trotzdem: C.H. Tings Anregungen vor Augen, war es nicht mehr allzu schwer, sich die nötigen Floating-Point-Assembler-Unterlagen zu beschaffen und "Druck-fehler" auszubügeln.
Ich habe das Programm von C.H.Ting umgearbeitet und ergänzt und dabei gleich eingedeutscht. Das ursprünglich in FD Jan/97 veröffentlichte Programm hatte folgende teilweise schwerwiegenden Fehler:

Im Hauptblock auf Seite 16 fehlt in den Zeilen 11 bis 27 das definierende Wort FPU .

(FINIT) wird in FINIT verwendet. Soweit, so gut. In FINIT wird aber der Gleitkomma-Stack mit dem Wert 0 vollgeschrieben. Das ist alles andere als "Initialisierung". (FINIT) kennzeichnet alle Gleitkomma-Stack-Einträge als "leer". Das ist eine echte Initialisierung. (FINIT) reicht also vollauf. Ich habe daraus dann wieder FINIT gemacht.

Das DO in FLOAT führt dazu, daß sich das System bei eingegebenen Zahlen der Form xx., wie in den Zeilen 19, 30 und 51 auf Seite 18, in der 64K langen Schleife verfängt. Ein ?DO klärt die Situation.

Das 3E in FRSTOR hat mich sehr viel Zeit gekostet. Es muß 26 heißen.

FSAVE und FRSTOR muß mit HEX compiliert werden.

Im großen Block auf Seite 16 muß es heißen:

ED statt EE in Zeile 9
FA statt EA in Zeile 10
E1 statt CE in Zeile 14
D8 statt D9 in Zeile 18

Durch Anwendung von FINIT werden alle FStack-Zellen geleert. Gibt man dann z.B. F+ ein, so entsteht eine "NaN". Das Wort F. nach C.H.Ting hängt sich dann auf. Gibt man 4.5 FLOAT 0.0 FLOAT F/ ein, so entsteht eine "positive Unendlichkeit". F. hängt sich dann auf. Gibt man -4.5 FLOAT 0.0 FLOAT F/ ein, so entsteht neine "negative Unendlichkeit". F. hängt sich dann auf. Ich habe F. entsprechend erweitert.

Zuerst wird über FXAM das TOS abgefragt und die erwähnten Sonderfälle erfahren eine Sonderbehandlung. Erst wenn sicher ist, daß es sich um eine "normalisierte Zahl" handelt, wird in das ursprüngliche F. aus dem Programm von C.H. Ting gegangen.
Hier meine Version des Programms von C.H.Ting,

ausgearbeitet für, und ausprobiert mit, Turbo-Forth: Die Befehle sind ANS-konform. Nicht alle ANS-Befehle sind eingearbeitet. Einige Befehle gehen über den ANS-Wortschatz hinaus. Die Gleitkommazahlen werden in FBUFFER und im Gleitkomma-Stack in 80-Bit-Breite aufgenommen und abgelegt.

Ich habe ein paar Befehle zur Manipulation des Gleitkomma-Puffers FBUFFER hinzugefügt.

FLOAT benötigt zwischendurch zwei Plätze auf dem Gleitkomma-Stack für Umwandlungsarbeiten. So ist es zum Beispiel nicht möglich, den Gleitkomma-Stack per FLOAT vollzuschreiben. Man überzeuge sich davon über FDUMP. Mit FLD1 oder FLDPI oder dergleichen kann man den F-Stack dagegen sehr wohl vollschreiben. Jeder weitere Versuch, über achtmal FLD1 hinaus, erzeugt aber eine NaN (Not a Number) auf dem TOS. Das alles kann man über FDUMP nachprüfen.

Um FLOAT bei schon vollem Gleitkomma-Stack verwenden zu können, kann man st(7) und st(6) durch 6 FFREE 7 FFREE "leeren". Achtung: 0 FFREE "leert" zwar den TOS. Will man aber anschließend eine Zahl eingeben, z.B. über FLDPI, dann wird die leere Stelle nach unten geschoben (st(1)=leer) und die Eingabe darübergelegt.

FFREE-BOTTOM ist gleichbedeutend mit 7 FFREE .

```
BASE @
HEX
CREATE FBUFFER 80 ALLOT
2VARIABLE DTEMP

CODE FWAIT     ( -- , wartet bis Coprozessor fertig )
   9B C,       NEXT END-CODE

CODE FCHS ( F: r -- -r , ändere Vorzeichen )
   D9 C, E0 C,   NEXT END-CODE

CODE FABS ( F: r -- |r| , Absolutwert )
   D9 C, E1 C,   NEXT END-CODE

CODE FLDZ      ( F: -- 0.0 , lade 0.0 )
   D9 C, EE C,   NEXT END-CODE

CODE FLD1      ( F: -- 1.0  , lade 1.0 )
   D9 C, E8 C,   NEXT END-CODE

CODE FLDL2T    ( F: -- log2/10/ )
               ( , lade Log zur Basis 2 von 10)
   D9 C, E9 C,   NEXT END-CODE

CODE FLDL2E    ( F: -- log2/e/ )
               ( , lade Log zur Basis 2 von e)
   D9 C, EA C,   NEXT END-CODE

CODE FLDPI     ( F: -- pi , lade pi )
   D9 C, EB C,   NEXT END-CODE

CODE FLDLG2    ( F: -- log10/2/ )
               ( , lade Log zur Basis 10 von 2)
   D9 C, EC C,   NEXT END-CODE

CODE FLDLN2    ( F: -- loge/2/ )
               ( , lade Log zur Basis  e von 2)
   D9 C, ED C,   NEXT END-CODE

CODE FSQRT     ( F: r -- \/r , Quadratwurzel )
   D9 C, FA C,   NEXT END-CODE

CODE F+        ( F: r1 r2 -- r1+r2  , Addition )
   DE C, C1 C,   NEXT END-CODE

CODE F*        ( F: r1 r2 -- r1*r2 , Multiplikation )
   DE C, C9 C,   NEXT END-CODE

CODE F-        ( F: r1 r2 -- r1-r2 , Subtraktion )
   DE C, E9 C,   NEXT END-CODE

CODE F-R       ( F: r1 r2 -- r2-r1)
               ( , vertauschte Subtraktion )
   DE C, E1 C,   NEXT END-CODE

CODE F/        ( F: r1 r2 -- r1/r2 , Division )
   DE C, F9 C,   NEXT END-CODE

CODE F/R       ( F: r1 r2 -- r2/r1
               ( , vertauschte Division )
   DE C, F1 C,   NEXT END-CODE

CODE FDUP ( F:    r1 -- r1 r1 )
   D9 C, C0 C,   NEXT END-CODE       \ fld  st(0)

CODE FDROP     ( F:     r -- )
   D8 C, D8 C,   NEXT END-CODE

CODE FSWAP     ( F: r1 r2 -- r2 r1 )
   D9 C, C9 C,   NEXT END-CODE       \ fxch st(1)

CODE FOVER     ( F: r1 r2 -- r1 r2 r1 )
   D9 C, C1 C, NEXT END-CODE

CODE FTUCK     ( F: r1 r2 -- r2 r1 r2 )
   D9 C, C9 C,                       \ fxch st(1)
   D9 C, C9 C,   NEXT END-CODE       \ fld  st(2)

CODE FROT ( F: r1 r2 r3 -- r2 r3 r1 )
   D9 C, C9 C,                       \ fxch st(1)
   D9 C, CA C,   NEXT END-CODE       \ fxch st(2)

CODE F-ROT     ( F: r1 r2 r3 -- r3 r1 r2 )
   D9 C, CA C,                       \ fxch st(2)
   D9 C, C9 C,   NEXT END-CODE       \ fxch st(1)

CODE FCOS      ( F:     r -- cos[r] )
   D9 C, FF C,   NEXT END-CODE

CODE FSIN      ( F:     r -- sin[r] )
   D9 C, FE C,   NEXT END-CODE
```

```
CODE FSINCOS ( F:    r -- cos sin )
    D9 C, FB C,   NEXT END-CODE

CODE FPTAN    ( F:     r -- y x , y/x=tan[r] )
    D9 C, F2 C,   NEXT END-CODE

CODE FPATAN  ( F:   y x -- r , r=arctan[y/x] )
    D9 C, F2 C,   NEXT END-CODE

CODE FSCALE  ( F:  r1 r2 -- r1 r2*2**r1
                   ( , scale r2 by 2**r1 )
    D9 C, FD C,   NEXT END-CODE

CODE FINIT      (  --  , F-Stack löschen )
    DB C, E3 C,   NEXT END-CODE

CODE FCLEX     (  --  , Exception-Flags löschen )
    DB C, E2 C,   NEXT END-CODE

CODE FADDR    ( F:  r1 r2 --; -- r1+r2 )
    DE C, C1 C,   NEXT END-CODE

CODE F2XM1    ( F:      r -- [2**r]-1 )
    D9 C, F0 C,   NEXT END-CODE

CODE FYL2X    ( F:  r1 r2 -- r1*[ld[r2]] )
    D9 C, F1 C,   NEXT END-CODE

CODE FYL2XP1 ( F: r1 r2 -- r1*[ld[r2+1]] )
    D9 C, F9 C,   NEXT END-CODE

CODE FXTRACT ( F: r -- e m
    ( , Umkehrung von FSCALE, Binärdarstellung ! )
    D9 C, F4 C,   NEXT END-CODE

CODE FNOP     (  -- , keine Wirkung    )
    D9 C, D0 C,   NEXT END-CODE

CODE FEXAM    ( F:  --; -- n   , FTOS untersuchen )
    D9 C, E5 C,                    \ FXAM
    DF C, E0 C,                    \ FSTSW AX
    AX PUSH      NEXT END-CODE

CODE FFREE-BOTTOM   ( F: --; -- , st[7] leeren     )
    DD C, C7 C,  NEXT END-CODE
' FFREE-BOTTOM 3 + DUP

: FFREE       ( F:  --; n -- , st[n] leeren )
    DUP 0 7 BETWEEN NOT
    IF ." Unzulässiger Eingabewert" DROP EXIT THEN
    C0 + LITERAL C! FFREE-BOTTOM
    C7   LITERAL C! [ 0. ] ; 2DROP

\ Verschiedene Funktionen

CODE F0<    ( F: r -- r ; -- flag ) \ st(0) < 0 ?
    FCLEX              \ Flags zurücksetzen
    D9 C,  E4 C,       \ FTST
    DF C,  E0 C,       \ FSTSW AX
    4100 # AX AND      \ Flagregister auslesen
    100 # AX SUB       \ F0< ?
    0=
    IF -1 # AX MOV
    ELSE AX AX XOR
    THEN
    AX PUSH
    NEXT END-CODE

CODE F<      ( F: r1 r2 -- r1 r2; -- flag ) \ st(0) < st(1) ?
    FCLEX              \ Flags zurücksetzen
    D8 C,  D1 C,       \ FCOM ST (1)
    DF C,  E0 C,       \ FSTSW AX
    4100 # AX AND      \ Flagregister auslesen
     100 # AX CMP      \ F< ?
    0=
    IF -1 # AX MOV
    ELSE AX AX XOR
    THEN
    AX PUSH
    NEXT END-CODE

CODE F>      ( F: r1 r2 -- r1 r2; -- flag ) \ st(0) > st(1) ?
    FCLEX              \ Flags zurücksetzen
    D8 C,  D1 C,       \ FCOM ST (1)
    DF C,  E0 C,       \ FSTSW AX
    4100 # AX AND      \ Flagregister auslesen
    0=
    IF -1 # AX MOV
    ELSE AX AX XOR
    THEN
    FSWAP              \ F: r1 r2
    AX PUSH
    NEXT END-CODE

CODE F>=     ( F: r1 r2 -- r1 r2; -- flag )
    FCLEX              \ Flags zurücksetzen
    D8 C,  D1 C,       \ FCOM ST (1)
    DF C,  E0 C,       \ FSTSW AX
    4100 # AX AND      \ Flagregister auslesen
     100 # AX SUB      \ F< ?
    0=
    IF     AX AX XOR
    ELSE -1 # AX MOV
    THEN
    AX PUSH
    NEXT END-CODE

CODE F<=  ( F: r1 r2 -- r1 r2; -- flag ) \ st(0) <= st(1) ?
    FCLEX              \ Flags zurücksetzen
    D8 C,  D1 C,       \ FCOM ST (1)
    DF C,  E0 C,       \ FSTSW AX
    4700 # AX AND      \ Flagregister auslesen
    0=
    IF     AX AX XOR
    ELSE -1 # AX MOV
    THEN
    AX PUSH
    NEXT END-CODE

CODE F0=     ( F: r -- r; -- flag )    \ st(0) = 0 ?
    FCLEX              \ Flags zurücksetzen
    D9 C,  E4 C,       \ FTST
    DF C,  E0 C,       \ FSTSW AX
```

```
    4100 # AX AND       \ Flagregister auslesen
    4000 # AX SUB          \ F0= ?
    0=
    IF -1 # AX MOV
    ELSE AX AX XOR
    THEN
    AX PUSH
    NEXT END-CODE

CODE F=     ( F: r1 r2 -- r1 r2; -- flag ) \ st(0) = st(1) ?
    FCLEX                  \ Flags zurücksetzen
    D8 C,  D1 C,           \ FCOM ST (1)
    DF C,  E0 C,           \ FSTSW AX
    4100 # AX AND          \ Flagregister auslesen
    4000 # AX CMP          \ F= ?
    0=
    IF -1 # AX MOV
    ELSE AX AX XOR
    THEN
    FSWAP                  \ F: r1 r2
    AX PUSH
    NEXT END-CODE

CODE F@ ( F: -- r; a -- )     \ Intel temp-real 80 bits
    BX POP
    DB C,  2F C,           \ FLD 0 [BX]
    NEXT END-CODE

CODE F! ( F: r -- ; a -- ) \ Intel temp-real 80 bits
    BX POP
    DB C,  3F C,           \ FSTP 0 [BX]
    NEXT END-CODE

CODE (D>F) ( F: -- r )
    DB C,  06 C,           \ FILD long-integer
    DTEMP ,
    NEXT END-CODE

: D>F ( d -- ; F: -- r )
    SWAP                   \ Bytes vertauschen
    DTEMP 2! (D>F) ;

CODE (F>D) ( F: r -- )
    DB C,  16 C,           \ FIST long-integer
    DTEMP ,
    NEXT END-CODE

: F>D ( -- d ; F: r -- )
    (F>D) DTEMP 2@
    SWAP ;                 \ Bytes vertauschen

\ Gleitkomma-Eingabe und -Ausgabe

DECIMAL
VARIABLE FSIGN
VARIABLE FEXPONENT
VARIABLE FDIGITS

: PRECISION        ( -- n ) FDIGITS @ ;
: SET-PRECISION    ( n -- ) FDIGITS ! ;
  5 SET-PRECISION

: JUSTIFY     ( F: 10.0 f1 -- 10.0 f2 , 1.0 <= f2 < 10.0 )
    F0< IF FABS -1 ELSE 0 THEN FSIGN !
    0 FEXPONENT !
    F<
    IF
      BEGIN
        FOVER F* F<
      WHILE
        -1 FEXPONENT +!
      REPEAT
      FOVER F/
    ELSE
      BEGIN
        FOVER F/ F<
        1 FEXPONENT +!
      UNTIL
    THEN ;

: F. ( F: f -- )
    FEXAM 18176 AND            \ 4700h and
    DUP 16640 AND 16640 =      \ 4100h; C3 and C0=1?
    IF ." leer"       DROP FDROP EXIT THEN
    DUP  1280 =                \  500h;
    IF ." +unendlich" DROP FDROP EXIT THEN
    DUP  1792 =                \  700h;
    IF ." -unendlich" DROP FDROP EXIT THEN
    DUP   256 =                \  100h;
    IF ." +NaN"       DROP FDROP EXIT THEN
    DUP   768 =                \  300h;
    IF ." -NaN"       DROP FDROP EXIT THEN
    DUP 16384 =                \ 4000h;
    IF ." +0.0"       DROP FDROP EXIT THEN
    DUP 16896 =                \ 4200h;
    IF ." -0.0"       DROP FDROP EXIT THEN
        512 OR 1536 =
    IF
      10. D>F FSWAP
      JUSTIFY
      FSIGN @
      IF ." -" ELSE SPACE THEN
      FDIGITS @ 0 DO FOVER F* LOOP
      F>D <# FDIGITS @ 0 DO # LOOP ASCII . HOLD
      #S #> TYPE
      FEXPONENT @ ?DUP
      IF ." E" 1 .R THEN
      FDROP FDROP FDROP EXIT
    THEN
    ." denormalisiert?" FDROP ;

: FLOAT   ( d --; F: -- f )
    10. D>F  \ 0.001234, 1234.5678 als Eingabe Basis
    D>F
    DPL @ 0
    ?DO FOVER F/ LOOP \ Auch z.B.  12345.  zulässig.
    FSWAP FDROP ;

HEX

CODE FSAVE     ( -- )
    DD C,  36 C,       \ FSAVE FBUFFER
    FBUFFER ,
    9B C,              \ FWAIT: Coprozessor fertig ?
    NEXT END-CODE
```

```
CODE FRSTOR  ( -- )
   DD C,  26 C,       \ FRSTOR FBUFFER
   FBUFFER ,
   9B C,              \ FWAIT: Coprozessor fertig ?
   NEXT END-CODE

DECIMAL

: FDUMP      ( -- )
   FSAVE              \ FStack --> FBUFFER
   FBUFFER            \ FStack dann leer
   14 +
   8 0
   DO DUP F@ F.   \ Aus FBUFFER holen und anzeigen
    10 +              \ Nächste 8-Byte-Einheit im FBUFFER
   LOOP
   FRSTOR  \ FBUFFER --> FStack
   DROP ;

: FB<F@  ( -- ) FSAVE FRSTOR ;
           \ Kopiere FStack nach FBUFFER

: FB>F!    ( -- ) FRSTOR ;
           \ Kopiere FBUFFER nach F-Stack

: FB<TOS@   ( F: r -- ; n -- )      \ Kopiere TOS nach
   DUP 0 7 BETWEEN NOT    \ Eingang n von FBUFFER
   IF
    ." Ungültiger Eingabewert"
    DROP EXIT
   THEN
   10 * FBUFFER 14 + + F! ;

: FB>TOS! ( F: -- r ; n -- )        \ kopiere Eingang n
                                    \ von FBUFFER
    DUP 0 7 BETWEEN NOT   \ nach TOS
   IF
    ." Ungültiger Eingabewert"
    DROP EXIT
   THEN
   10 * FBUFFER 14 + + F@ ;

BASE !
```

Professionelle Schrifteffekte mit Corel Draw

Jeder der schon einmal damit beschäftigt war, Vorlagen für eine Veranstaltung herzustellen, kennt das Problem. Was ist wohl die treffendste grafische Aufbereitung für die angestrebte Veranstaltung ? Stunden vergehen, „tausend“ unterschiedliche Motive werden ausprobiert, verworfen, verändert, wieder aktualisiert usw.. Dann endlich ist es soweit, das treffendste Motiv wurde erstellt oder gefunden. Jetzt nur noch die W - Fragen : Wer ? Wann ? Wo ? Wieviel ? usw. in der passenden Schriftgröße plaziert, fertig !

Aber Moment - da war doch noch etwas. Genau, die Schrift ! Diese grundlegende Information gerät immer mehr in den Hintergrund.„ Professionelle Schrifteffekte mit Corel Draw !“, bietet eine neue Möglichkeit sich mit diesem Thema zu beschäftigen.
Die Schritt für Schritt Anleitungen zur Gestaltung auffälliger Schrifteffekte mit Corel Draw ab Version 4, rückt das gedruckte Wort wieder in den Mittelpunkt unserer Aufmerksamkeit.

Die auf einer CD mitgelieferten Working Models von Corel Draw 5 und 6, Photo Paint, Correl Cadd und den Corel Xara Graphics File Viewer, bieten das Rüstzeug für die typographische Reise auf Gutenbergs Spuren.
Die Gestaltung von Schriftzügen wie Knetmasse, Neonlicht, luftigen Seidentüchern oder Metall, dimensionale Verformungen, Auto - Stereogramme, oder eigener Hompages u.v. m, sind leicht aufbereitete Lerninhalte.

Aber auch der Blick hinter die Kulissen des typographischen Handwerks wird ermöglicht. Dazu zählen geschichtliches Hintergrundwissen ebenso wie eine praktische Einführung in Perspektive und Schattenkonstruktion. Nicht nur Schriften, sondern auch das Wissen um ihre Anwendung für die unterschiedlichsten Vorlagen, zeigt nach der Lektüre des Buches Tiefenwirkung. Die Reiseleiter Günter Cürten und Michael Ziegert überzeugen durch ihren spielerischen Umgang mit dem Thema, ohne fachliche Informationen in den Hintergrund zu stellen. Gönnen sie sich ein Aha - Erlebnis, man gönnt sich ja sonst nichts.

HANSER
ISBN 3-446-18806-1
79,- DM
incl. CD-ROM
1996

Hinweis: Nutzer von ZF, F-PC und WIN32FOR müssen das FSAVE umbenennen. In diesen Systemen ist FSAVE bereits mit einer Funktion belegt, die das jeweils aktuelle System, einschließlich der durch den Nutzer definierten Worte, auf den Massenspeicher sichert. Fred Behringer schlägt vor, als “Ersatz” das Wort **FSTOR** zu verwenden.

EuroFORTH '97
26-28 September
St. Anne's College
Oxford

Die 13. EuroFORTH-Konferenz fand dieses Mal Ende September in Oxford statt. Untergebracht war die Conferenz in St. Anne's College, einem relativ jungen College, das bis vor kurzem ein reines Mdchen-College war. Man darf sich die Universittsstadt Oxford nicht so vorstellen, wie deutsche Universitätsstädte: Statt einer zentral verwalteten Universität gibt es zahllose Colleges, die alle von Stiftern (adeligen, kirchlichen, privaten) gegründet wurden. Zu jedem College gehören Gebäude zur Unterbringung der Studenten, Vorlesungsäsle (``Theaters''), ein Speisesaal und bei kirchlichen Colleges natürlich noch eine Kirche.

Nach dem Mittagessen ging es dann am Freitag los. Das Motto der Konferenz war ``Embedded Communications'', und in der Tat gab es sogar drei Vorträge zu diesem Thema (Änderungen im Tagungsablauf lasse ich jetzt erst einmal außen vor.) :

N. J. Nelson stellte in seinem Vortrag ``A tale of three Forths'' das Netzwerk fr die Steuerung einer Wäscherei vor. Traditionell wird so etwas mit einem PLC (Programmable Logic Controller) gemacht; die Verdrahtungskosten sind dabei immens (ca. 20 km Kabel). Deshalb wird stattdessen ein ``virtueller PLC'' verwendet, der über ein Feldbusprotokoll mit den Sensoren und Aktoren kommuniziert. Das erste Forth ist dabei das Forth im Feldbuskontroller IX1 von Delta-T. Das eigentliche Feldbus-Protokoll wird hier in Software verarbeitet, was größtmögliche Flexibilität gibt (man ist nicht von vornherein an ein Protokoll gebunden); Fehler in der Protokoll-Implementierung können viel einfacher behoben werden. Das zweite Forth läuft auf einem RTX2001 und implementiert den PLC. Das dritte Forth ist das Interface im PC. Es spricht mit dem virtuellen PLC über ein VxD unter Windows 95. Das einzige Teil des Systems, das nicht in Forth geschrieben ist, ist eben jenes VxD. Der Autor sucht einen Forth-VxD-Compiler; meiner Meinung nach sollte er sich lieber ein anderes Betriebssystem suchen (Linux etwa).

Steven Coul, *Graham Stevenson* und *Stephen Pelc* präsentierten den PowerNet UDP/IP-Stack. Es implementiert ein Subset des Internet-Protokolls, also nur IP(Internet Protocol) , UDP(User Datagram Protocol) und ICMP(Internet Control Message Protocol) . Da eine serielle Schnittstelle weiter verbreitet ist als ein Ethernet-Adapter, wird das etwas veraltete SLIP (Serial Line Internet Protocol) verwendet. Der Programmierer verwendet ein Subset des BSD Socket Datagram interface, oder KEY, KEY?, EMIT, TYPE. In Zukunft soll noch das wichtigste Internet-Protokoll, TCP(Transmittion Control Protocol) implementiert werden. Ein embedded Web-Server und ein einfaches FTP (File Transfer Protocol) sollen ebenfalls hinzukommen.

A. J. Willox und *D. R. H. Bags* stellen in ihrem Vortrag ``Distributed Data Acquisition and Processing Systems for Industry, Education and Training'' ein CAN(Controller Area Network) - Bus-basiertes System zur verteilten Datenerfassung und Steuerung vor.

Die nächste Sitzung beschäftigte sich mit ``Languages and Notation''.

Wolf Wejgaard zeigte in seinem Vortrag ``Objects in HolonForth'' seine Version des objekt-orientierten Forths. Bei ihm gibt es keine Klassenhierarchie; die Methodentabelle muß für jede Klasse vollständig beschrieben werden. Damit wird auch die Implementierung sehr kompakt.
Er verwendet ein “message object”-Modell, wobei die Message eine globale Variable setzt, die dann vom Objekt ausgewertet wird. Ein “object message”-Modell könnte auf dieses unschöne Detail verzichten.

Auch *Dr. Sergey A. Sidorov's* Vortrag “Data in DSSP - prefix access in postfix language” ging in die Richtung “Objektorientiertes Forth”. Die Basis-Datentypen verstehen zwar alle Messages, und haben geeignete Methoden dafü implementiert; allerdings kann man keine neuen Messages definieren. Neue Datentypen dagegen lassen sich definieren. Dieses Modell ist ebenfalls “message object”.
Offenbar ist diese Unsitte sehr weit verbreitet.

Peter Knaggs schweifte mit dem Thema “Perl vs. Forth” etwas vom Tagungsthema ab. Er leitete aber aus den Features von Perl neue Forderungen an Forth ab: Es sollte endlich ein standardisiertes Objekt-Modell verwendet werden, und damit soll eine String-Library zur Verfügung gestellt werden, die Perls Funktionalität enthält.
Anton Ertl warf ein, daß die Standardisierung des Objekt-Modells kein Problem ist, wenn die Library zur Verfügung steht, und auch weit benutzt wird.

J. H. Morrish beschrieb in seinem Vortrag "Rapid development of real time multi-sequence control programmes" eine etwas andere Variante des Multi-Taskings. Seine Programme sind eigentlich State machines, die ihren Status in ein paar wenigen Variablen abspeichern, beim Aufruf über CASE- und IF-Anweisungen Aktionen ausführen und den Nachfolge-Status berechnen.

Am Freitag Abend war dann noch "Punting" angesagt. Dabei stakt man ein Boot durch einen kleinen Fluß in Oxford. Auch wenn das die Briten für eine typisch britische Aktivität halten, ist diese Art der Fortbewegung in Fernost weit verbreitet. Peter Knaggs ging dabei über Bord und mußte deshalb nicht für Spott sorgen.

Nach dem anschließenden Bar-BQ zeigen sich noch die Tücken eines Mädchen-College: Die Hintertür ist abgeschlossen, und auch nicht von innen zu öffnen (die Mauer dagegen für junge Männer keineswegs unüberwindlich). Immerhin ist die Vordertür noch offen. Anschließend wird noch bis tief in die Nacht debattiert.

Die dritte Session "Applications" begann mit **Stephen Pelc** und "A Portable Open Software Architecture for Industry". Es geht dabei um die SENDIT VM, eine virtuelle Maschine, die stark an Forth erinnert, und für die Europay Open Terminal Architecture (OTA) verwendet wird.
Anders als OpenFirmware handelt es sich bei SENDIT tatsächlich um compilierten Code, nicht nur tokenisierte Source. Auaerdem ist SENDIT sprachunabhängig, d.h. man kann auch C nach SENDIT compilieren.

Auch der Vortrag "TIDE: Expoiting Forth in a Windows Environment" beschäftigte sich mit SENDIT, bzw. dem Debuggen von OTAs. Es stellt sowohl die Simulation von den Terminals und deren möglichen Peripheriegeräten zur Verfügung, als auch Tracing und das Setzen von Breakpoints; wie man es eben von Forth gewohnt ist. Die Bedienoberflächen der Terminals werden als Windows-Dialoge simuliert.

Paul Frenger schnitt mit "Forth and Artificial Vision" ein gänzlich anderes Thema an. Es ging um ein künstliches Auge. Nicht als Prothese für einen Menschen, sondern natürlich für einen Roboter. Er verwendet eine kleine CCD-Kamera, die ein NTSC-Signal ausgibt; dieses kann dann von einem Frame-Grabber digitalisiert werden.

Auch *Duncan Louttit* bot mit seinem "`Real-time maze solving" etwas zum Anfassen an. Er beschreibt nicht nur seine elektronische "Maus", und wie sie ihren Weg durch ein Labyrinth findet, sondern hat auch eine Demonstration und ein Video da. Ebenfalls zeigt er, was die kritischen Punkte sind. Die Maus muß einen schnellen Prozessor haben, damit sie das Labyrinth auch in der geforderten Zeit erforschen und lösen kann, und sie muß auch gute Bodenhaftung und einen starken Motor haben, damit nicht ihre mechanische Geschwindigkeit der bremsende Faktor ist.

Die vierte Session befaßte sich mit "Techniques".

Anton Ertl und *Christian Pirker* berichten mit "The Structure of a Forth Native Code Compiler" den aktuellen Stand des RAFTS-Projekt. Hier wird der Forth-Source zuerst in einen Datenflußgraphen verwandelt, auf den dann moderne Compiler-Techniken wie Scheduling und Register-Allocation losgelassen werden können. Außerdem werden Wörter inline compiliert, also nicht über einen Call aufgerufen. Das eröffnet weitere Optimierungsmglichkeiten, vor allem eine bessere Belegung der Register. Der zusätzliche Aufwand beim Compilieren wird größtenteils durch die höhere Geschwindigkeit des erzeugten Codes kompensiert - damit wird ja auch der Compiler schneller. Allerdings bleibt für weitere Optimierungen noch Luft.

Adin Teven stellte in "Taming the Trampoline" eine Variante von lokalen Variablen vor. Er benennt Stack-Elemente. Das Aufräumen des Stacks erledigt er durch Umkopieren der Elemente. Er vergleicht zwar in seinem Vortrag verschiedene Varianten mit lokalen Variablen, bleibt aber den Vergleich mit dem optimierten, direkten Stackzugriff schuldig.

Karl-Dietrich Neubert zeigte mit "Flash-Sort: Sorting by in situ Permutation" einen O(n)-Sortieralgorithmus. Dabei wird die Permutation berechnet, mit der das unsortierte Array in ein sortiertes verwandelt werden kann. Entscheident für die niedrige Ordnung des Algorithmus ist aber die Klassenbildung. Eigentlich ist die Sortierung dann O(n*m), wobei m die Länge der Daten ist.

Egmont Woitzel und *Stephan Lange* stellten, wie schon auf der Forth-Tagung '97, mit dem Paper "STDCALL Threaded Code and its Impact on Debugging" ein STDCALL-Threaded Forth vor. Dieses Schema erlaubt es, Win32-DLLs direkt mit Forth zu

verknüpfen, ohne den Umweg über Wrapper-Funktionen zu gehen. Da es in ihrer Implementierung ein zentrales NEXT gibt, kann man durch einfaches Auswechseln des NEXTs sehr einfach Debugging-Funktionen implementieren.

In der füften Session ging es um “User Interface and Useability”.

Dmitry V. Frantov und *Mikhail N. Shumakov* zeigten in ihrem Vortrag “DED (DSSP Editor+Debugger)”, wie man ihr Forth-ähnliches DSSP-System debuggt und editiert. Es handelt sich dabei, ähnlich wie Holon, um ein hypertextartiges System, dessen Entwicklungsumgebung in “Development system” und “Target system” aufgeteilt wird, wobei die Aufteilung nur notwendig ist, wenn das Programm zu groß ist, um mitsamt Debugger in den DOS-Speicher zu passen.

Matt Purvis und *Stephen Pelc* beschrieben in “Interfacing to the Windows 95 Common Controls”, wie sie den Kampf mit den täglich (Oder doch nur alle paar Jahre?) neuen Features in Windows aufnehmen. Den Kampf gegen die Common Controls haben sie überzeugend gewonnen, lassen aber die Drachen ActiveX und COM noch in ihren Höhlen, auch wenn daran schon gearbeitet wird.
Nachdem ich den Kampf mit dem LCD-Projektor (der schon Wolf Weygaard Probleme bereitete) gewonnen hatte, konnte ich auch “MINOS - Visual bigFORTH” vorführen. MINOS löst das beklagte Problem, daß mit Forth graphische Benutzeroberflächen nicht so elegant programmiert werden können wie mit Visual BASIC oder Delphi. Zwar existiert inzwischen eine Portierung auf Windows 95/NT, aufgrund der Instabilität dieser Plattform und der mangelnden Motivation des Autors ist Linux nach wie vor die Plattform der Wahl. MINOS kommt mit einer eigenen Widget-Library, und umgeht so das Problem, mit unöntig komplexen Bibliotheken und immer neuen Standards zu kämpfen.
Am Abend findet dann ein “Formal Dinner” statt.
In Session 6 “Miscellaneous” kommen alle die Themen dran, die anderswo keinen Platz gefunden haben.

Howerd Oakford wagte mit “Forth: Past, Present and Future” einen Blick in die Zukunft. Das Hauptproblem bei Forth sieht er als Kommunikationsproblem: “Wie verteile ich meinen Code”. Versionskontrolle scheint dazu ein unabdingbares Mittel, um in den zu schaffenden Libraries den Überblick zu wahren.

Peter Knaggs forderte “A truly International Standard” und sprach damit das Problem der Internationalisierung an. Zu diesem Thema wurde dann auch ein Workshop ins Leben gerufen, dem inzwischen auch eine Mailingliste (international-forth@mips.complang.tuwien.ac.at) folgte. Peter stellt Forderungen zu Datums- und Geld-Formaten, Zeichensätze, Sortierungsreihenfolge und ähnliches werden dann im Work shop angesprochen.
Anton Ertl schlägt vor, hier vom C-Standard abzugucken, und eventuell fehlende Sachen etwa Java nachzumachen.

John D. Carpender stellte seinen “Fuzzy Interpreter” vor. Der kann Datenströme mittels Fuzzy Logic auswerten, und dabei bestimmte Muster erkennen.

Ewald Pfau und sein Vortrag “A landscape and its map” zeigten Verbindungen von Forth und der bildenen Kunst (wenn ich das richtig verstanden habe).
Die letzte Session befaßt sich mit Forth und “Critical Systems”

Paul E. Bennett beschäftigte sich mit “Forth in Safety Critical Systems/Configuration and Certification” vor allem mit der Zertifizierung von Software. Er vertritt dabei den Standpunkt, daa Forth eine der wenigen Sprachen ist, die eine Zertifizierung überhaupt ermöglichen (siehe auch Ullrich Hoffmanns Vortrag auf einer der vergangenen Forth-Tagungen). Der hierarchische Aufbau von Forth-Programmen ermöglicht es, jedes Teil sauber zu beschreiben, und damit die Korrektheit relativ einfach sicherzustellen.

Michael Milendorf zeigte mit “The Error Reporting and Handling Solution in OpenFirmware Selftest Methods”, wie man komfortable und aussagekräftige Fehlermeldungen erzeugt.

Malcolm Bugler schlug mit “FORTH in Critical Care Environments” in dieselbe Kerbe wie Paul. Er schlägt “risk management” als Strategie zur Reduzierung von Fehlern vor. Hierzu wird die Matrix aus Fehlerhäufigkeit und der zu erwartende Effekt in drei Zonen eingeteilt: “Intolerable”, “ALARP (as low as reasonable possible)” und “broadly accepted”. Die nicht tolerierbaren Fehler müssen selbstverständlich auf 0 zurückgeführt werden, die Fehler im ALARP-Bereich anschließend minimiert werden.
Daß Forth für kritische Aufgaben ideal geeignet ist, wird auch hier auf die Testbarkeit und die besondere Struktur von Forth-Programmen zurückgeführt.

Bernd Paysan

Gehaltvolles

zusammengestellt und übertragen
von Fred Behringer

Forth Dimensions der Forth Interest Group, USA

Juli/August 1997

7 A Platform-Independent Token System for Payment Terminals
Peter Johannes, Stephen Pelc, and Elizabeth Rather

Die Finanzwelt bemüht sich eifrig, neue Instrumente zu installieren, die das Finanzgebahren des einzelnen revolutionieren sollen. Intelligente Karten, Kreditkarten, elektronische Geldbörsen und wie die Reizworte alle heißen mögen, die mit ihren erahnbaren Folgen gewisse Kreise elektrisieren. Hardware-Beschränkungen bilden einen wesentlichen Teil der sich ergebenden Probleme. Wie soll man Überweisungs-Software in das bißchen RAM packen, das in eine Plastikkarte herkömmlicher Abmessung paßt? Und die soll dann auch noch einen Mikroprozessor mit Ein/Ausgabe-Funktionen enthalten. Forth bietet sich in ganz natürlicher Weise bei Vorliegen solcher Beschränkungen an und es stellt sich heraus, daß es auch diesmal wieder, bei der Entwicklung der angesprochenen Technologie, eine führende Rolle spielt.

12 A Simple Implementation of the Kermit Protocol in Pygmy Forth
Frank Sergeant

Frank Sergeant, der Anhänger unter denjenigen hat, die Forth "rank und schlank" sehen wollen, brachte im letzten Heft eine Beschreibung seiner Kermit-Implementation. Hier nun folgt das Programm.

37 Yet Another Forth Objects Package
Anton Ertl

Programmierer sind häufig mit der Aufgabe beschäftigt, verschiedene Datenstrukturen in der einen Hinsicht gleich, in einer anderen Hinsicht unterschiedlich behandeln zu müssen. Eine umfangreiche CASE-Struktur wäre nicht sehr elegant und müßte gewartet werden. Seinem Wesen nach ist das ein Problem, das von objektorientierten Systemen gelöst wird. Nach seinen kritischen Bemerkungen über das Neon-Modell im letzten Heft liefert der Autor heute ein Modell, das er für besser hält, und geht auf dessen Implementierung ein.

Het Vijgeblaadje 6 der HCC FORTH-gebruikersgroep, Niederlande
Oktober 1997

De Forth bibliotheek (2)
Roelf Toxopus

Ein weiterer Bericht über Forth-Bücher, die Mitglieder der HCC-Forth-Gebruikersgroep kostenlos entleihen können. Am Ende des Artikels werden 14 vorhandene Titel aufgeführt. Ich gebe im folgenden 2 daraus an, die mir irgendwie weniger bekannt vorkommen:
Kelly&Spies, Forth: a text and reference, 487 S., Prentice Hall 1986; F79, F83 .
Olney&Benson, Fundamental Forth, 239 S., Pan Books 1985; Fig Forth, F79, F83 .

(POSTZ)EGELS wg
Leendert van den Heuvel

Es besteht eine Arbeitgruppe, die sich mit der AT89Cx051-Entwicklungsumgebung beschäftigt. Natürlich in Forth programmiert. Es wird ein Überblick über weitere Vorhaben gegeben.

Het Vijgeblaadje 7 der HCC FORTH-gebruikersgroep, Niederlande
November 1997

Forth-gg produkten

AT89C2051-CPU, 15,- Gulden
C64 Prä-ANS-Forth, 15,- Gulden; 16-Bit-beinahe-ANS-Forth auf Floppy

Gehaltvolles

Ultrasoon afstandmeter
Willem Ouwerkerk

Ultraschall-Abstandsmesser, passive Infrarot-Bewegungs-sensoren, AT89C2051, programmiert in ByteForth.

Schudwoord
Leendert van den Heuvel

Ein LED-Laufschrift-Simulationsprogramm in Forth. Sinusförmige Hin-und-Herbewegung, ein Zwischending zwischen Laufschrift-Matrix und Nipkov-Scheibe.

Forth Dimensions der Forth Interest Group, USA

September/Oktober 1997

7 Writing a Macintosh Application with Pokket Forth
Ronald T. Kneusel

Forth bringt einen Hauch von gesundem Menschenverstand in die Welt der GUI-Entwicklung. Der Autor behauptet nicht, Mac-Experte zu sein, stellt aber ereignis-gesteuerte (event-driven) Programmierung mit diesem beschnittenen System vor. Wie man sehen wird, nimmt ein vertrautes Werkzeug manchen Fluch von dem, was Forth-Programmierer so lange vermieden haben.

13 Yet Another Forth Structures Package
Anton Ertl

Im letzten Heft hat der Autor ein Modell für ein objektorientiertes Forth vorgeschlagen und auf den vorliegenden Artikel hingewiesen. Das heute vorzustellende Programmpaket bietet Unterstützung für solche Elemente wie STRUCT aus C oder RECORD aus Pascal. Enthalten sind auch eine automatische Wortausrichtung (alignment) und die Optimierung von Feldern mit Offset 0.

17 Approaching CREATE DOES>
Dave Taliaferro

Das Definieren von definierenden Worten braucht für den neuen Forth-Programmierer kein Schreckgespenst zu bleiben. Im Gegenteil, es kann als Mittel der Erleuchtung betrachtet werden, um zu einem tieferen Verständnis der Sprache und zu einem wirkungsvolleren Umgang mit ihr zu gelangen. Der Autor schrieb diesen Artikel mit der Vorstellung, anderen zu helfen, während er sich selbst noch gar nicht so weit vom Lernprozeß entfernt und die Dinge, die ihm Schwierigkeiten bereiteten, immer noch vor Augen hat.

22 Lookup Tables
Hans Bezemer and Benjamin Hoyt

Okay, sie sind wirklich kein Glanzstück und man kann sie kaum elegant nennen. Aber die Autoren legen überzeugend dar, daß Lookup-Tables (Sprungtabellen) höchst nützlich, flexibel, leicht zu warten, in vielen Richtungen hin erweiterbar und wirklich die richtige Lösung für viele Probleme sind. Und mit den vorzuschlagenden Werkzeugen (tools) wird ihre Implementation kinderleicht.

37 Pygmy Embellishments
Richard W. Fergus

Wenn man jahrelang ein bestimmtes Forth-System verwendet hat, hat man sich nicht nur Übung verschafft. Ziemlich sicher hat man sich auch ein Paket von Hilfswerkzeugen geschaffen, die für bestimmte Anwendungen, zur Beseitigung erkannter Schwachpunkte nach eigenem Geschmack entwickelt wurden. Ergreifen Sie die Gelegenheit und erkunden Sie die persönliche Werkzeugkiste (the personal toolkit) eines Pygmy-Anhängers.

Vorsicht Humor

Das Microsoft-Windows-95-Entwicklungsteam ist auf Betriebsausflug in Irland. Sie mieten einen Jeep und fahren übers Land. Zwischendurch werden sie aufgehalten, da eine Schafherde die Straße kreuzt. Daraufhin kommen sie mit dem Schäfer ins Gespräch. Bill Gates will mit dem Schäfer um ein Schaf wetten, daß er den Beruf des Schäfers erraten kann. Danach darf der Schäfer versuchen, den Beruf des Teams zu erraten. Schafft er es, gehört ihm der Jeep. Der Schäfer ist einverstanden. Darauf sagt Bill Gates: Sie sind Schäfer. Genau, antwortet der Schäfer. Daraufhin gehen zwei Microsoftmitarbeiter in die Herde und holen sich ein Tier, das sogleich geschlachtet, gebraten und verzehrt wird. Nun ist der Schäfer dran: Sie sind Entwickler von Windows 95. Bill Gates ist entsetzt. Er zückt den Autoschlüssel und fragt: Woher wußten sie das? Der Schäfer: Das war leicht. Nur Windows-95-Entwickler nehmen aus einer Herde von 300 Schafen den Hund mit. (mk)

Von Klaus Kohl gesammelt aus:
Markt&Technik, # 43, 24/10.97

"Ich habe mir erlaubt, Windows 95 von meiner Festplatte zu entfernen." Arthur Clarke, Schöpfer von '2001: Odyssee im Weltraum', darüber, wie der erste Satz des sprechenden Computers HAL gelautet haben könnte.

Von Birgit Prinz gesammelt aus:
NRZ, 1997

"Kerniges" zum WIN32FOR

Bei Arbeiten zur "Entrümpelung" am Kernel von WIN32FOR bin ich unter anderem auf die folgenden Definitionen gestoßen:

```
: UPC( char -- char )
  \ convert char to uppercase
  SP@ REL>ABS 1 SWAP
  uppercase_x XCALL DROP ;

: UPPER  addr len -- )
  \ convert string addr,len
  \ to uppercase
  SWAP REL>ABS
  uppercase_x XCALL DROP ;

: LOWER  addr len -- )
  \ convert string addr,len
  \ to lowercase
  SWAP REL>ABS
  lowercase_x XCALL DROP ;
```

Das FORTH übergibt eine Arbeit, die es selbst viel schneller tun könnte, freiwillig an WINDOWS ? Die Übergabe erfolgt noch nicht einmal "direkt", sondern läuft erst noch durch einen der xcalls (Externe Calls) des Wrappers ? Da blutet dem FORTHer das Herz. Das kann so nicht bleiben, jedenfalls nicht in einem System mit dem ich meine ohnehin knapp bemessene Freizeit vernichte. Seit einigen Tagen stehen darum, nach einer Metakompilation des Kernels, folgende Definitionen im Wörterbuch:

```
CODE UPC ( chr -- CHR )
  \ Zeichen auf dem Stack in Gross-
  \ buchstaben umwandeln
  XOR EBX, # 32
     NEXT C;
```

Das gab es vorher noch gar nicht in TOM's System. Aber weil es gar zu einfach war, habe ich das Gegenstück zu UPC gleich mit definiert.

```
CODE LOC ( CHR -- chr )
  \ Zeichen auf dem Stack in Klein-
  \ buchstaben umwandeln
  OR EBX, # 32
    NEXT C;

CODE UPPER ( adr len -- )
  \ String in Grossbuchstaben
  \ umwandeln
    MOV ECX, EBX \ ECX = len
    POP EBX      \ len von TOS
        \ nehmen. [EBX] = adr
@@1: CMP 0 [EBX] [EDI], # 228 \ ae ?
     JE SHORT @@2
    CMP 0 [EBX] [EDI], # 252 \ ue ?
     JE SHORT @@2
    CMP 0 [EBX] [EDI], # 246 \ oe ?
     JE SHORT @@2
    CMP 0 [EBX] [EDI], # 97  \ <a ?
     JL SHORT @@3
    CMP 0 [EBX] [EDI], # 122 \ >z ?
     JG SHORT @@3
@@2: XOR 0 [EBX] [EDI], # 32
@@3: INC EBX
   LOOP @@1
    POP EBX  \ adr vom TOS nehmen
       NEXT C;

CODE LOWER ( adr len -- )
    MOV ECX, EBX
    POP EBX
@@1: CMP 0 [EBX] [EDI], # 196 \ AE ?
     JE SHORT @@2
    CMP 0 [EBX] [EDI], # 220 \ UE ?
     JE SHORT @@2
    CMP 0 [EBX] [EDI], # 214 \ OE ?
     JE SHORT @@2
    CMP 0 [EBX] [EDI], # 65  \ <A ?
     JL SHORT @@3
    CMP 0 [EBX] [EDI], # 90  \ >Z ?
     JG SHORT @@3
@@2:  OR 0 [EBX] [EDI], # 32
@@3: INC EBX
   LOOP @@1
    POP EBX
       NEXT C;
```

Selbstverständlich muß UPPERCASE ebenfalls etwas direkter arbeiten. Das kann diese Version...

```
: UPPERCASE ( str -- STR )
   DUP COUNT UPPER
  ;
```

Meine Definition von UPPER arbeitet auf meinem 83 MHz Pentium Overdrive ~ 14 Mal schneller als die originale Version dieses Wortes. Möglicherweise lassen sich UPPER und LOWER noch schneller definieren. Ich bin mit Jim Schneider's Assembler nicht sehr vertraut. In jedem Fall hat sich aber der Aufwand, den ich hier treiben mußte, gelohnt. Immerhin mag man mir zustimmen, wenn ich diesen Vergleich als Beweis für die häufig zu hörende Behauptung werte, daß WINDOWS unverhaeltnismäßig viel Resourcen frißt, auch Resourcen der CPU.

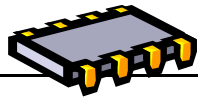
Und außerdem ist es mir so einfach angenehmer ! **;-)**

Friederich Prinz

```
\ ein File um zwei Fernthermometer mit PWS zu protokollieren       M.Bitter 8.97
\ verwendet werden soll der Sensor SMT160-30-92.

\ Stichworte: ZF; FASTGRAF

empty

\ zuerst die grafischen Darstellungen

fload f:\fastgraf.seq                   \ Kernroutinen laden
fload f:\fgpcx.seq                      \ Routinen fuer PCX-Bilder
fload f:\fgmouse.seq
fload i:\forth\samples\disassem.seq
fload help                              \ eigene Hilfedatei

warning on                              \ Protest bei Mehrfachdefinition

hidden also         \ Worte dieses Vocabulars sichtbar machen

: init                                  \ der TSR-Treiber FGDRIVER.EXE
  " fgdriver " bl "syscommand ;         \ wird geladen

0 Constant vga_modus            \ Platzhalter fuer den Bildschirmmodus
0 Constant Seiten               \ ... fuer die Anzahl an Bildschirmen im Speicher

comment:
 AUFLOESUNG emittelt die maximale Anzahl von Bildschirmen, die im Speicher gehal
 ten werden koennen und meldet den entsprechenden Bild schirmmodus bei FASTGRAF
 an.
comment;

: aufloesung ( -- )             \ "beste" Aufloesung ermitteln
  60 fgsvgainit drop            \ vor fgbestmode aufrufen ( Autodetect)
  3 0 DO 640 480 4 I - dup      \ bis zu maximal vier (I) Seiten
         =: Seiten fgbestmode dup \ anfordern
         =: vga_modus           \ merken
         -1 = not               \ war es ein gueltiger Wert?
         IF leave THEN          \ FALLS JA --> Erfolgreich fertig
      LOOP                      \ ansonsten alles nochmal
  vga_modus -1 =                \ hatten wir Erfolg?
  IF
  abort" Die Grafikkarte konnte nicht initialisiert werden?"
  THEN
  vga_modus fgsetmode ;         \ ermittelten Modus setzen

\ Die Pfadangaben beziehen sich auf mein System.

Create bildpfad ," f:\fenstr_l.pcx" here 30 allot 30 erase

0 0 2Constant Ecke      \ Platz fuer zwei Koordinaten (oben, links)

: >ecke ( n n -- )      \ lies: zur Ecke
  ['] Ecke >body 2! ;   \ speichert zwei Koordinaten

: links ( -- )          \ Koordinaten des linken Bildes (Instrument)
  20 0 >ecke ;

: rechts ( -- )         \ Koordinaten des rechten Bildes (Instrument)
  380 0 >ecke ;


: einblenden ( -- )           \ zeigt ein PCX-bild an Position x y
  fgmove                      \ Grafikcursorposition setzen
  bildpfad 1+ %10 fgshowpcx   \ Bild laden und zeigen (%10=Bitmaske Ladeoptionen)
  drop ;                      \ Rueckgabeflag ignorieren
```

```
: instrumente ( -- )                \ zeigt zwei Messinstrumente an
  " f:\fenstr_l.pcx" bildpfad place \ Bilddatei (um)benennen
  links ecke einblenden             \ linke Ecke als Bezugspunkt; Bild anzeigen
  " f:\fenstr_r.pcx" bildpfad place \ bilddatei (um)benennen
  rechts ecke einblenden ;          \ linke Ecke als Bezugspunkt; Bild anzeigen

comment:
 Nun werden die Bilder der leeren Displays bzw. die Bilder der einzelnen Ziffern
 eingeblendet. Dazu werden die verschiedenen Koordinaten als Konstanten oder in
 einer Tabelle gehalten. Die PCX Dateien der Ziffern enthalten in ihrem Namen die
 jeweilige Ziffer, diese Ziffer wird in den Pfadstring gepatcht und so die pas-
 sende Datei geladen.
comment;

48  Constant display_x          \ relative X-Koordinate des Displays
116 Constant display_y          \ relative Y-Koordinate des Displays
25  Constant im_pfad            \ an Position 25 im Filepfad steht die
                                \ jeweilige Ziffer ( 7seg_x.pcx)

Create positionen               \ x Koordinate fuer div. Ziffernpositionen
  54 dup , 21 + dup , 15 + dup , 21 + dup , 18 + dup , 4 + ,

: leer ( -- )                    \ blendet ein leeres Display ein
  " f:\display.pcx" bildpfad place   \ Bilddatei (um)benennen
  ecke                               \ Bezugskoordinaten
  display_y + swap display_x + swap  \ absolute Koordinaten bilden
 einblenden ;                        \ Bild anzeigen

: Ziffer ( n pos -- )              \ Ziffer n an Position rel zu x y schreiben
  swap                             \ Stackschieberei
  " f:\7seg_0.pcx" bildpfad place  \ Bilddatei (um)benennen
  Ascii 0 +                        \ Zahl in Ziffer wandeln
  bildpfad im_pfad + c!            \ in den Bildpfad patchen
  2* positionen + @                \ rel. Position holen
  ecke                             \ Bezugskoordinaten
  display_y + 40 + -rot            \ absolute Y-Position
  + swap                           \ absolute x-Position, richtige Reihenfolge
  einblenden ;                     \ Bild anzeigen

: leerstelle ( pos -- )            \ schreibt eine Leerstelle an Position pos
  " f:\7seg_bl.pcx" bildpfad place \ Bilddatei (um)benennen
  2* positionen + @                \ rel. X-Korrdinaten
  ecke                             \ Bezugskoordinaten
  display_y + 40 + -rot            \ absolute Y-Koordinaten
  + swap                           \ absolute x-Position, richtige Reihenfolge
  einblenden ;                     \ Bild anzeigen

: plus/minus ( -- )               \ Bild an Vorzeichenposition ausgeben
  ecke                            \ Bezugskoordinaten
  display_y + 40 + swap           \ absolute y-Koordinate
  positionen @ + swap             \ absolute x-Koordinate
  einblenden ;                    \ Bild anzeigen

: plus ( -- )                      \ Pluszeichen ausgeben
  " f:\7seg_bl.pcx" bildpfad place \ Bilddatei (um)benennen
  plus/minus ;                     \ Leerstelle an Vorzeichenposition

: minus ( -- )                      \ Minuszeichen ausgeben
  " f:\7segsign.pcx" bildpfad place \ Bilddatei (um)benennen
  plus/minus ;                      \ Minuszeichen an Vorzeichenposition

: punkt ( -- )                      \ schreibe Dezimalpunkt
  " f:\7seg_dot.pcx" bildpfad place \ Bilddatei (um)benennen
  ecke                              \ Bezugskoordinaten
  display_y + 40 + 29 + swap        \ absolute Y-Koordinaten bilden
  4 2* positionen + @  + swap       \ absolute X_Koordinaten bilden
  einblenden ;                      \ Bild anzeigen
```

```
: .display  ( n -- )                \ zeigt n im Display an (/10el Grad)
  dup 0< IF minus ELSE plus THEN        \ passendes Vorzeichen ausgeben
  abs 10 /mod 10 /mod 10 /mod drop      \ drei Ziffern erzeugen
  2 ziffer 3 ziffer 5 ziffer punkt ;    \ Ziffern und Dez.punkt schreiben

comment:
 Um den optischen Eindruck zu beurteilen hier ein Testwort TEST_TH,
 das zwei Anzeigeinstrumente auf dem Bildschirm ausgibt und mit den
 Pfeiltasten bedienbar ist.
 Mit der Pfeil-rauf- bzw. Pfeil-runter-Taste wird die "Temperatur"
 in 1/10-tel Gradschritten veraendert, mit den Tasten Pfeil-rechts
 und Pfeil-links wird das jeweilige Messinstrument eingeschaltet.
 Abgebrochen wird mit der ESCAPE-Taste.
comment;

: test_th ( -- )                    \ zeigt zwei Messinstrumente an
  init                              \ den Fastgraf TSR-Treiber anmelden
  aufloesung                        \ Fastgraf Bildschirmmodus setzen
  instrumente                       \ rechtes und linkes Messinstrument anzeigen
  rechts                            \ das rechte Instr. ist das aktive
  0 dup .display                    \ Null anzeigen; eine Null bleibt TOS
  Begin key dup $18 = IF swap 1- dup .display swap THEN   \ 1/10el Grad runter
            dup   5 = IF swap 1+ dup .display swap THEN   \ 1/10el Grad rauf
            dup   4 = IF rechts oer .display THEN         \ rechtes Instrument
            dup $13 = IF links  over .display THEN        \ linkes Instrument
        $1b =                       \ Abbruch
  UNTIL drop                        \ fertig; Stack aufrumen
  dark ;                            \ zurueck in den Textmodus

comment:
 so: Die Grafik ist fertig! Das alles geht natuerlich eleganter mit Bildern im
 (virtuellen) Videospeicher usw. aber die Zeit, mir da den Durchblick zu holen,
 habe ich im Moment nicht.
comment;
```



... von der Stirne heiß, rinnen muss der Schweiß?

Temperaturabfrage und Darstellung mit ZF

von Martin Bitter
Möllenkampweg 1a, 46499 Hamminkeln,

Stichworte: Bitmaps, Fastgraf, ZF

Spätsommer:

Es ist ein heißer Tag im Spätsommer. Die großen Ferien sind schon einige Zeit vorbei. die durchgehende Fensterfront der Klasse zeigt nach Südost, die Sonne brennt mit voller Kraft.
An den verschiedenen Gruppentischen herrschen unterschiedliche Meinungen darüber, was sinnvoll ist: Fenster gekippt, Jalousien schräggestellt --> Durchzug mit Sonnenschein; oder Jalousien hochgestellt --> Schatten ohne Durchzug. Subjektiv sind kaum Unterschiede festzustellen. Alle, SchülerInnen und Lehrer, leiden unter den Temperaturen.

Ein Anlass!

Das ist doch einer dieser häufig gewünschten, aber doch recht seltenen “echten” Lernanlässe: Einmal durch Ausprobieren, Nachmessen und Vergleichen herauszubekommen, welche die sinnvollste Maßnahme ist, den Klassenraum kühl zu halten.

Doch um das umzusetzen, bräuchte man: zwei Thermometer und eine Möglichkeit diese Abzulesen, ohne die Messsituation wesentlich zu beeinflussen, schön wäre eine halbautomatische Protokollierung.

Das sieht doch nach einer Aufgabenstellung für eine Messstation aus - wenn dann noch einer der Lehrer FORTHkundig ist, dann haben wir (endlich) ein Problem: Sie wissen ja: FORTH ist die Lösung.

Mitgefühl - mit Gefühl

Es gibt viele Möglichkeiten, Temperaturmesswerte zu bestimmen und in den Computer zu bekommen. Etwas Stöbern in Versandkatalogen führte mich zu dem Temperatursensor SMT 160-30-92. Dieser Sensor arbeitet mit VSS = 4,75 - 7,5 V= und liefert ein Rechtecksignal, dessen Tast**verhältnis** der jeweiligen Temperatur entspricht. Also ein (theoretisch) recht simpel zu handhabender Sensor.

Um einen Messwert zu erhalten, wird in dem Programm das Statusregister des Parallelanschlusses in vier Endlosschleifen abgefragt und der gelesene Wert jeweils mit der zugehörigen Bitmaske eines Pins (hier 10 und 14) verglichen.
Die ersten zwei Abfragen dienen dazu, einen Signalwechsel (Flanke) möglichst genau zu erkennen, in den beiden folgenden Abfragen werden die Zeitlängen des Highpegels und des Lowpegels ermittelt und zur weiteren Verwendung auf den Stack gelegt.

```
$378 Constant LPT1              \ Basisadresse LPT1
LPT1 1 + Constant Statusreg     \ Adresse Statusreg.

Code Status> ( -- n )                  \ Statusregister lesen
    Statusreg #    DX mov
                   0 AL in
 %11111000 #       AX and
                   AX push
next end-code
```

```
: roh_temperatur ( -- n n )                \ liest Tasverhältnis
      BEGIN status> Sensor @ and 0= UNTIL   \ auf low_pegel warten
      BEGIN status> Sensor @ and 0<> UNTIL  \ auf high_pegel warten
      Zaehler @ 0 zaehler !                 \ Zähler lesen und setzen     -- identisch -+
      BEGIN zaehler incr                    \ Zähler um eins erhöhen                    wegen
      status> Sensor @ and 0= UNTIL         \ auf low_pegel warten                      Zeitver-

      Zaehler @ 0 zaehler !                 \ Zähler auslesen, initialisieren       halten
      BEGIN zaehler incr                    \ Zähler um eins erhöhen                    |
      Status> Sensor @ and 0<> UNTIL        \ auf Hig-Pegel warten          ------------+
      Zaehler @ rot drop ;                  \ Zähler auslesen; sinnlosen Wert vernichten!
```

In die Augen - in den Sinn

Das Programm soll ein Anlass sein, Temperaturwerte festzuhalten und zu beobachten, es soll nicht eine automatische Analyse leisten. Es ist darum “sinnig” die Messergebnisse ständig vor

den Augen zu haben. Wichtig ist mir eine gefällige Darstellung der Messwerte.

Unter den Möglichkeiten, die die vorhandene Harware bietet (386er, VGA, MS-Dos) scheidet Windows aus - besonders wegen meiner "Unmöglichkeit" unter Windows ungestört die parallele Schnittstelle abzufragen.

Jawohl Herr (Fast-)Graf

Letztendlich entschied ich mich, die Temperaturwerte durch die Abbilder zweier digitaler Messgeräte darzustellen.

Das programmtechnische Mittel der Wahl war die von F. Prinz für ZF implementierte Fastgraf Bibliothek.
Fastgraf ist eine professionelle Sammlung von Grafikroutinen, die als TSR-Programm im (DOS)-Speicher liegen und über einen Interrupt angesprungen werden können. Fastgraph 4.03 (lite) in der vorliegenden Version meldet sich beim ersten Aufruf mit einer Copyrightmeldung in einer käuflich zu erwerbenden Vollversion unterbleibt diese.

Die Bilder des Messinstrumentes habe ich aus der Katalog-CD eines Versandhandels per Screenshot kopiert, und für die Darstellung auf dem VGA-Standardschirm leicht überarbeitet (in der Höhe gekürzt).
Die einzelnen Ziffern und Zeichen sowie das leere Display wurden mit einem herkömmlichen Malprogramm gestaltet.

Im (hoffentlich) ausreichend kommentierten Listing ist zu sehen, wie mit den Fastgrafroutinen *.PXC-Dateien eingelesen und ausgegeben werden können. Es wäre natürlich möglich, alle benötigten Bitmaps im Speicher zu halten, aber aus Zeitgründen habe ich darauf verzichtet, zudem erwies sich der Festplattencache als so groß, dass alle Bilder in ihm vorgehalten wurden, die häufigen Lesezugriffe zur Aktualisierung der Zahlausgabe also durch (schnelle) virtuelle Festplattenzugriffe erfolgen.

Tempus fugit

Das ganze Programm und die Hardwareaustestung erfolgte an einem Wochende und gelang mit ZF und FASTGRAF zu meiner Zufriedenheit, die meiste Zeit benötigte der Versand der zwei Sensoren.

Zum Einsatz gelangte das Programm (noch) nicht: einer der Sensoren ging wegen unbekannter Pinbelegung und genauso ungeschicktem wie risikofreudigen Bastler (M.B.) "über den Jordan", bis Ersatz beschafft war, waren die heißen Tage (glücklicherweise) vorbei.
Aber der nächste Sommer kommt bestimmt!

Ergänzung

Der Programmcode zur Abfrage der Thermometer ist ein wenig komplexer als hier gezeigt: Es gibt eine Sicherheitsabfrage (sind Thermometer angeschlossen?), die Kernroutinen sind in Maschinencode gehalten und die Umrechnung Tastverhältnis-Temperaturwert erfolgt mit einer Tabelle, da die (erlesenen) Werte nicht ausreichend linear waren.

(Für Tips bin ich dankbar!)

Martin Bitter

Borland Delphi 3

für Einsteiger und Fortgeschrittene

Das Buch aus dem HANSER Verlag hält, was der Titel verspricht. In drei Teilen und mehr als 700 Seiten erfährt der geneigte Leser alles, was er zur Arbeit mit Delphi 3 benötigt. Im Grundlagenteil wird er nach einer Einführung in die Programmierung unter Windows in die Grundlagen von Object Pascal eingeführt. Der Praxisteil des Buches wartet mit vielen Tips und Tricks zu Standardproblemen auf und vier Beispielapplikationen vermitteln mehr als nur eine Übersicht darüber "wie Profis es angehen". Dem Prinzip "so viel wie nötig" folgend, bietet das Buch auch dem versierten Programmierer Antworten auf Fragen, die er in einschlägigen Dokumentationen vergeblich sucht. Alle Beispiele und Demos liegen auf einer CD bei.

HANSER Verlag
ISBN 3-446-19064-3
79,- DM

Borland Delphi 3
für Profis

Was im Grundlagenwerk "für Einsteiger und Fortgeschrittene" keinen Platz mehr fand, findet der Leser in diesem Buch: GDI-Grafikprogrammierung, Vektorgrafik, Registrierdatenbank, Fehlerbehandlung, dynamische Datenstrukturen, API-/DLL Programmierung, Komponentenentwicklung, ActiveX-/OLE Programmierung, Datenbanken, Reports, SQL, Netzwerk, Netzwerksicherheit, DDE, Dateien, Drucker, serielle Schnittstellen ... und vieles mehr. Praxis- und Applikationsteile runden das Werk ab. Auch der versierte Forther wird aus diesem Buch Nutzen ziehen, wenn es ihm zeigt, was sein "Forth für Windows" eigentlich alles können sollte... Dazu werden attraktive Tools, sowie alle Beispiele und Demos des Buches auf einer CD mitgeliefert.

HANSER Verlag
ISBN 3-446-19065-1
98,- DM

Forth-Gruppen regional

Rhein-Ruhr	Jörg Plewe Tel.: 0208-49 70 68 (p) Treffen: jeden 1. Samstag im Monat im S-Bahnhof Derendorf Münstererstraße 199.
Moers	Friederich Prinz Tel.: 02841-58398 (p) (Q) (Bitte den Anrufbeantworter nutzen !) Treffen: (fast) jeden Samstag, 14:00 Uhr, MALZ, Donaustraße 1 47443 Moers
Darmstadt	Andreas Söder Tel.: 06257-2744
Mannheim	Thomas Prinz Tel.: 06271-2830 (p) Ewald Rieger Tel.: 06239-8632 (p) Treffen: jeden 1. Mittwoch im Monat, Vereinslokal Segelverein Mannheim e.V., Flugplatz Mannheim-Neuostheim
München	Jens Wilke Tel.: 089-89 76 890 Treffen: jeden 4. Mittwoch im Monat, China Restaurant XIANG Morungerstraße 8 München-Parsing

µP-Controller Verleih

Thomas Prinz
Tel.: 06271-2830 (p)

Gruppengründungen, Kontakte

Regional	Stuttgart Wolf-Helge Neumann Tel.: 0711-8 87 26 38 (p)
Fachbezogen	8051 ... (Forth statt Basic, e-Forth) Thomas Prinz Tel.: 06271-2830 (p)

Forth-Hilfe für Ratsuchende

Forth allgemein	Jörg Plewe Tel.: 0208-49 70 68 (p)
	Karl Schroer Tel.: 02845-2 89 51 (p)
	Jörg Staben Tel.: 02103-24 06 09 (p)

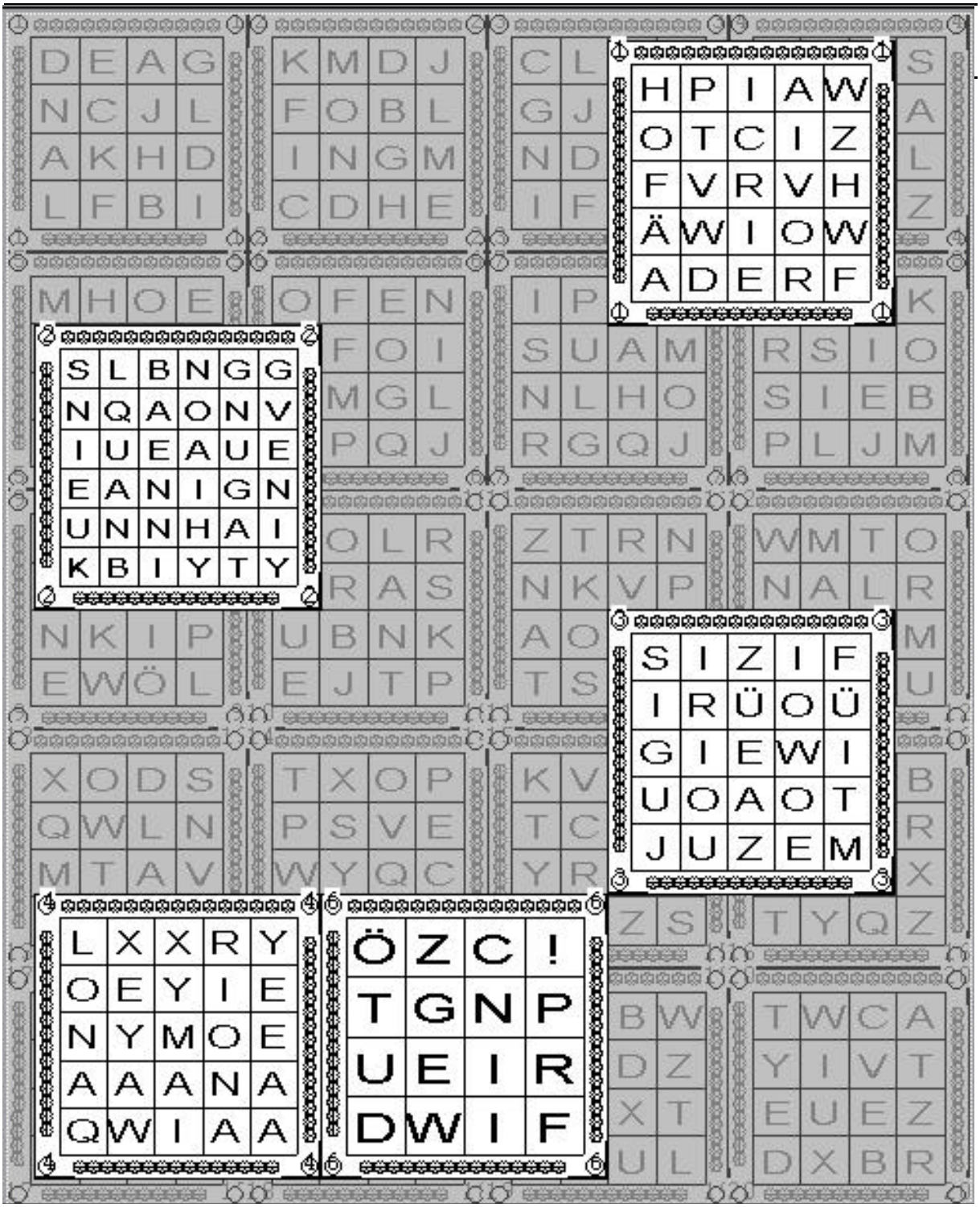
Spezielle Fachgebiete

Arbeitsgruppe MARC4	Rafael Deliano Tel./Fax: 089-841 83 17 (p)
Anfänger und Wiedereinsteiger	Gerd Limbach Tel.: 02051-25 51 12 (p)
32FORTH (Atari)	Rainer Aumiller Tel.: 089-6 70 83 55 (g) (p)
FORTHchips (FRP 1600, RTX, Novix)	Klaus Schleisiek-Kern Tel.: 040-375 008 03 (g)
F-PC & TCOM, Asyst, emb.Contr., Fuzzy...	Arndt Klingelnberg Tel.: 02404-6 16 48 (p) (g) (Q)
HS/Forth (Harvard Softworks)	Wigand Gawenda Tel.: 030-44 69 41 (p)
KI, Object Oriented Forth, Sicherheitskritische Systeme	Ulrich Hoffmann Tel.: 04351 -712 217 (p) Fax: -712 216
Forth-Vertrieb volksFORTH / ultraFORTH RTX / FG / Super8 / KK-FORTH	Ingenieurbüro Klaus Kohl Tel.: 08233-3 05 24 (p) Fax : 08233-99 71
Forth-Mailbox (KBBS)	0431-533 98 98 (8 N 1) Sysop Holger Petersen hp@kbbs.org Tel.: 0431-533 98 96 (p) bis 22:00 Fax : 0431-533 98 97

Hinweise zu den Angaben nach den Telefonnummern:

Q = Anrufbeantworter
p = privat, außerhalb typischer Arbeistzeiten
g = geschäftlich

Die Adressen des Büros der Forthgesellschaft und der VD finden Sie im Impressum des Heftes.



"SUCHSEL" von Martin Bitter
Auflösung im nächsten Heft

Erkennen Sie die Botschaft ?