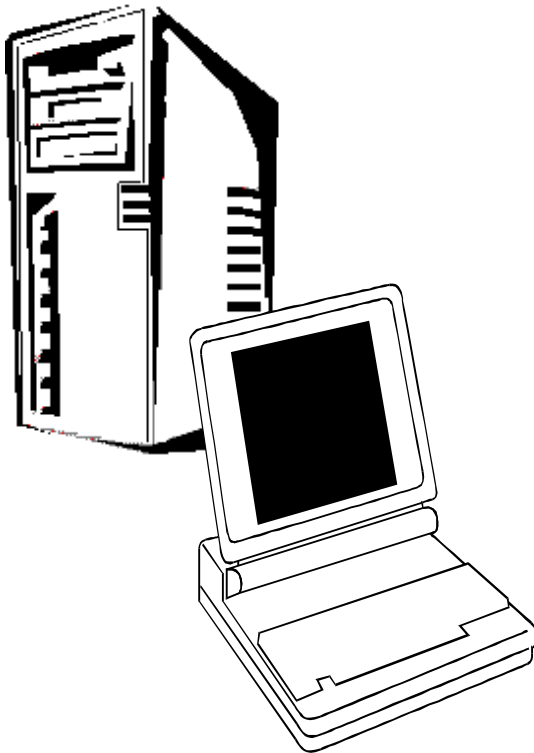
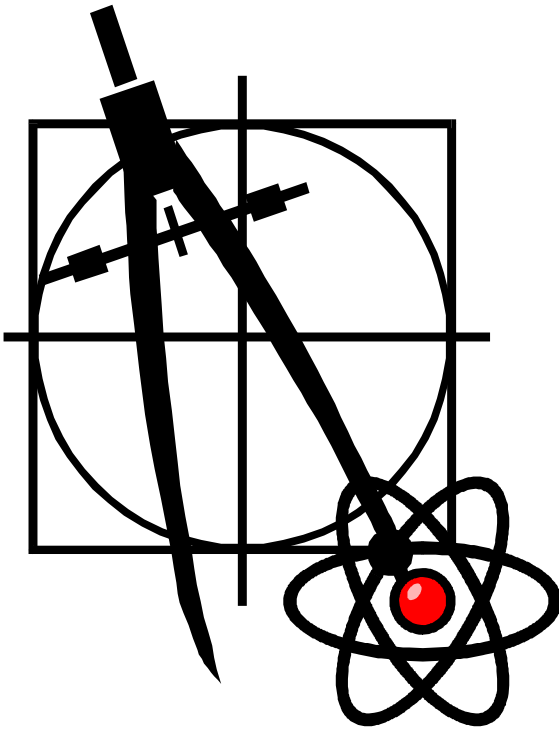


für Wissenschaft und Technik, für kommerzielle EDV,  
für MSR-Technik, für den interessierten Hobbyisten.



### In dieser Ausgabe

#### Leserbrief

Ein Leser schreibt, was ihn interessiert

#### ... arithmetisierte Logik

damit kann man rechnen? Logisch!

#### Web-Forth

ein interaktives Forth im Browserfenster

#### Reorder

wenn zwei ..., freut sich der dritte

#### Gehaltvolles

aus den Schwesterzeitschriften

#### Ein Mino $\Sigma$ Beispiel:

ein GUI für OpenSched

#### Berichte

über die Jahrestagung 2001 der  
Forth Gesellschaft e.V.

#### Das Protokoll

der Jahrestagung 2001 der  
Forth Gesellschaft e.V.

## Dienstleistungen und Produkte fördernder Mitglieder des Vereins

### tematik GmbH Technische Informatik

Feldstrasse 143  
D-22880 Wedel  
Fon 04103 - 808989 - 0  
Fax 04103 - 808989 - 9  
mail@tematik.de  
www.tematik.de

Gegründet 1985 als Partnerinstitut der FH-Wedel beschäftigten wir uns in den letzten Jahren vorwiegend mit Industrieelektronik und Präzisionsmesstechnik und bauen z.Z. eine eigene Produktpalette auf.

Know-How Schwerpunkte liegen in den Bereichen Industriewaagen SWA & SWW, Differential-Dosierwaagen, DMS-Messverstärker, 68000 und 68HC11 Prozessoren, Sigma-Delta A/D. Wir programmieren in Pascal, C und Forth auf SwiftX68k und seit kurzem mit Holon11 und MPE IRTC für Atmel AVR.

### Hier könnte IHRE Anzeige stehen

**Setzen Sie sich doch einfach einmal mit dem Büro der Forthgesellschaft e.V. in Verbindung.**

### Dipl.-Ing. Arndt Klingenberg

Tel.: ++32 +87 -63 09 89 (Fax: -63 09 88)  
Waldring 23, B-4730 Hauset, Belgien  
akg@aachen.kbbs.org

Computergestützte Meßtechnik und Qualitätskontrolle, Fuzzy, Datalogger, Elektroakustik (HiFi), MusiCassette HighSpeedDuplicating, Tonband, (engl.) Dokumentationen und Bedienungsanleitungen

### Achtung:

**Unvorhergesehene Änderung der Redaktions-E-Mail-Adresse!**

**Adresse(n)**

**VD@FORTH-EV.DE**

**[Martin.Bitter@t-online.de](mailto:Martin.Bitter@t-online.de)**

**Näheres auf dem Rücktitel!**

### Forth Engineering Dr. Wolf Wejgaard

Tel.: +41 41 377 3774 - Fax: +41 41 377 4774  
Neuhöflirain 10  
CH-6045 Meggen <http://holonforth.com>

Wir konzentrieren uns auf Forschung und Weiterentwicklung des Forth-Prinzips und offerieren HolonForth, ein interaktives Forth Cross-Entwicklungssystem mit ungewöhnlichen Eigenschaften. HolonForth ist erhältlich für 80x86, 68HC11 und 68300 Zielprozessoren.

### 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 Entwicklungsbüro Dr.-Ing. Egmont Woitzel

Budapester Straße 80A 18057 Rostock  
Tel.: +49 (0381) 46139910 Fax: +49 (0381) 4583488

PC-basierte Forth-Entwicklungswerkzeuge, comFORTH für Windows und eingebettete und verteilte Systeme. Softwareentwicklung für Windows und Mikrocontroller mit Forth, C/C++, Delphi und Basic. Entwicklung von Gerätetreibern und Kommunikationssoftware für Windows 3.1, Windows95 und WindowsNT. Beratung zu Software-/Systementwurf. Mehr als 15 Jahre Erfahrung.

### 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.

Impressum	4
Editorial	4
Direktorial	5
<i>Ulrich Hoffmann</i>	
Neue Mitglieder	5
Leserbrief	7
<i>Fred Behringer</i>	
(Tagungsbeitrag) Lego-Roboter und arithmetisierte Logik in Forth	8
<i>Fred Behringer</i>	
Reorder und die Folgen	13
<i>Martin Bitter</i>	
(Tagungsbeitrag) WebForth	14
<i>Chris Jakeman</i>	
Gehaltvolles Teil 1	16
VIJGEBLAADJE Nr. 25	
<i>Fred Behringer</i>	
(Tagungsbeitrag) MINOΣ Example: OpenSched GUI	17
<i>Bernd Paysan</i>	
Buchbesprechungen:	23
Windows 2000 im professionellen Einsatz	
Technik der IP-Netze	
<i>Friederich Prinz</i>	
Forth öffnet die Türen	24
<i>Fred Behringer</i>	
Rechnen mit garantierter Genauigkeit	25
<i>Chrstitoph Pöppe</i>	
Gehaltvolles Teil 2	30
FORTHWRITE	
<i>Fred Behringer</i>	
Gehaltvolles Teil 3	31
VIJGEBLAADJE Nr. 26	
<i>Fred Behringer</i>	
Des Rätsels Lösung	32
- Zahlendarstellung	
Forthtagung 2001 in Hamminkeln-Dingden	34
<i>Friederich Prinz</i>	
Forthtagung 2001 in Hamminkeln-Dingden	35
<i>Bernd Paysan</i>	
Protokoll der Jahresversammlung 2001 der Forth-Gesellschaft e.V.	37
<i>Jens Wilke</i>	
Den Lego-Transmitter überlisten	40
- mit und ohne Forth	
<i>Fred Behringer</i>	
Verleihservice	41
<i>Thomas Prinz</i>	
From the big Teich	42
<i>Henry Vinerts</i>	

In der nächsten Vierten Dimension lesen Sie:

Micker-Forth	Wolfgang Allinger
Quartus	Wolfgang Allinger
Forth-Briefmarke	Hans Eckes
Karatsuba Teil 2	Martin Bitter
RCX-Tower-Hardware-Bastelei	Fred Behringer u. Martin Bitter



**IMPRESSUM**

**Name der Zeitschrift**

**Vierte Dimension**

**Herausgeberin**

Forth-Gesellschaft e.V.  
Postfach 16 12 04  
D-18025 Rostock  
Tel.(Anrufbeantw.): 0381-400 78 28  
Fax: 0381-400 78 28  
E-Mail:  
SECRETARY@FORTH-EV.DE  
DIREKTORIUM@FORTH-EV.DE

Bankverbindung: Postbank Hamburg  
BLZ 200 100 20  
Kto. 563 211 208

**Redaktion & Layout** (vorübergehend)

Martin Bitter  
Möllenkampweg 1a  
46499 Hamminkeln  
Tel.: 02857-1419  
E-Mail: VD@FORTH-EV.DE  
mbitter@bigfoot.de

**Anzeigenverwaltung**

Büro der Herausgeberin

**Redaktionsschluß 2001**

März, Juni, September, Dezember  
jeweils in der dritten 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, Aufbausketten u.ä., die zum Nichtfunktionieren oder eventuellem Schadhafwerden von Bauelementen oder Geräten führen, kann keine Haftung übernommen werden. Sämtliche Veröffentlichungen erfolgen ohne Berücksichtigung eines eventuellen Patentschutzes. Warennamen werden ohne Gewährleistung einer freien Verwendung benutzt.

**Vereinsmeierei und Glücksgefühle!**

Diese Ausgabe der Vierten Dimension enthält zu einem Teil Beiträge, die ursprünglich nicht für sie geschrieben waren. Es geht um Beiträge zur diesjährigen Tagung der Forth Gesellschaft e.V. Das könnte man als Vereinsmeierei abtun. Aber dem ist nicht so: Alle Beiträge bewegen sich auf hohem Niveau, sie sind es wert allen Vereinsmitgliedern zugänglich gemacht zu werden. Denn nicht jeder konnte die Tagung besuchen, wüsste aber dennoch gerne, was dort 'Sache war'. Und dazu ist die Vierte Dimension da: Den Austausch der forthbegeisterten und der Forthnutzer untereinander zu transportieren.

Aus Platzgründen können nicht alle Tagungsberichte in dieser Ausgabe abgedruckt werden. Einige werden mit der Herbstausgabe erscheinen.

Die Forth Gesellschaft ist ein kleiner Verein. Das ist in mancher Hinsicht schade. So ist die Absolutzahl der aktiven Mitglieder, das sind solche, die die Belange der Gesellschaft pflegen oder ihre Gedanken und Erkenntnisse in der Vierten Diskussion zur Disposition stellen zwar gering – prozentual gesehen aber wiederum recht hoch. Andererseits bringt diese Kleinheit auch Nähe, Familiarität und Geborgenheit. Erfahrungen und Gefühle, die auf jeder Jahrestagung vertieft werden. (Übrigens: Meiner Meinung nach ist der Mensch ein Gefühlswesen. Das! unterscheidet uns von den 'abstrakten Denkmaschinen', den kommenden „Giganten des Lichts“ (St. Lem). Erst wenn KI das begreift und umsetzt, müssen wir uns Sorgen um 'Rechenautomaten' als konkurrierende Lebensform machen.)

Seit der letzten Ausgabe der Vierten Dimension hatte ich zweimal in forthlichen Angelegenheiten Glücksgefühle:

Als Reaktion auf Ulrich Pauls Leserbrief (der Redakteur fleht: Leserbriefe, schickt Leserbriefe!) ergab sich ein reger Austausch von E-Mails und zum Schluss ein fertiges Produkt. Ich berichte darüber auf Seite 13.

Und der Drachenrat tagte, kreiste und gebar einen 'neuen' Drachenhüter: (Trara!!) Mich!

Ich habe jetzt die Aufgabe, diesen imposanten, schweren und nachtaktiven Kerl zu bewachen und zu hüten. Ich bin überzeugt, dass er nachts aktiv ist – das ist die einzige Erklärung für die Verwüstungen auf meinem Schreibtisch. Ich bin das nicht! So unordentlich kann gar kein Mensch sein!

Ich freue mich sehr, über die Anerkennung meiner Arbeit für die Forth Gesellschaft, glaube aber auch, dass andere den Drachen eher verdient haben. Nun ja! Rein statistisch gesehen steigt Jahr für Jahr für jedes Mitglied die Chance Drachenträger zu werden. Ich bin nur gespannt, wie die geheimen, obskuren Rituale des Drachenrates nächstes Jahr aussehen werden. Und ob es ein spezielles Drachenwasser für Antialkoholiker gibt :-)))



**Quelltext Service**

Die Quelltexte in der VD müssen Sie nicht abtippen. Sie können diese Texte auch direkt bei uns anfordern und sich zum Beispiel per E-Mail schicken lassen. Schreiben Sie dazu einfach eine E-Mail an die Redaktionsadresse.

MB



## Direktorial

Liebe Mitglieder der Forth-Gesellschaft,

die Mitgliederversammlung in Hamminkeln-Dingden hat mich am 29. April 2001 zum Nachfolger von Egmont Woitzel gewählt, der sein Direktoriumsamt aus beruflichen Gründen aufgeben musste. Zusätzlich habe ich auch die Position des WEB-Masters der Forth-Gesellschaft übernommen. Vielleicht erinnert sich ja noch der eine oder andere: Ich war bereits schon einmal Direktor, nämlich in den Jahren 1992-1994.

Auf der Jahrestagung habe ich den englisch-sprachigen Forth Wiki-Server vorgestellt; ein Projekt mit dessen Arbeit ich im Februar 2000 begonnen habe und der bisher auf einem kostenlosen Server im Internet erreichbar war. Ein Wiki-Server ist eine Sammlung von WEB-Seiten, die von jederman (und jeder Frau) direkt verändert und erweitert werden können. Der Forth Wiki-Server enthält WEB-Seiten rund um Forth. Während der Tagung entstand die Idee, den Forth Wiki-Server unter das Dach der Forth-Gesellschaft zu holen. Nun, der Umzug ist abgeschlossen und der Wiki-Server kann unter <http://www.forth-ev.de/wiki/> erreicht werden. Im gleichen Zuge hat er auch ein deutsch-sprachiges Gegenstück bekommen, auf dem wir Forth-Gesellschafts Dinge behandeln können. Ich hoffe, der Wiki-Server wird zur Datendrehscheibe unserer Aktivitäten und ergänzt so die Vierte Dimension, unsere WEB-Seiten, E-Mail, oder die Diskussionen in [de.comp.lang.forth](http://de.comp.lang.forth). Nur - von allein wird er das natürlich nicht. Also, nichts wie ran an die Tasten und mitgemacht.

Ein Höhepunkt der Jahrestagung war der Lego-Roboter-Wettbewerb. Außer viel Spaß hat er aber auch gezeigt, dass mehrere Lego-Roboter sich in einem Raum so ganz und gar nicht vertragen wollen, da sich ihre Infrarot-Übertragungen gegenseitig stören. Nun, was liegt näher, als sich dieses Problem genauer anzusehen. Auf Initiative von Michael Kalus entstand das LegoProtokoll-Projekt ("Lego zwitschert Forth") für das die Forth-Gesellschaft leihweise zwei Lego RCX-Prozessoren mit Infrarot-Übertragungstürmchen zur Verfügung stellt. Auf die Ergebnisse sind wir gespannt. Michael wird hier in der Vierten Dimension darüber berichten.

Ich denke, dies sind zwei gute Beispiele, wo sich spannende Dinge in der Forth-Gesellschaft tun. Ich kann nur jeden von Euch dazu einladen, an Projekten mitzumachen oder sie selbst ins Leben zu rufen. Das ist viel spannender als "nur" viermal im Jahr die VD zu bekommen.

Forthige Grüße,

Ulrich Hoffmann (uho@forth-ev.de)

P.S. Über einen Artikel für die VD freuen wir uns natürlich auch.

## Neue Mitglieder:

Im Namen des Direktoriums, meiner selbst und einiger weiterer mir persönlich bekannter Menschen begrüße ich herzlich folgende neue Mitglieder der Forth Gesellschaft e.V.

**Mario Deilmann** ist seit Anfang des Jahres Mitglied.. Herzlich willkommen!

Einiges aus dem Wirkungskbereich **Jörg Völkers** kann man bei [www.tematik.de](http://www.tematik.de) besichtigen. Dort erfährt man über seine Arbeiten an GraFidi (ich spreche das 'schwäbisch' aus), und Präzisionsmessinstrumenten, darunter Dosier- und Differentialwaagen, sowie dass er neben anderen Sprachen in Forth (Holon, Atmel) programmiert. Vgl. auch Anzeige auf Seite 2

**Dr. Klaus Beck** verdient unter anderem mit Forth seinen Lebensunterhalt. Wir hatten das Vergnügen ihn als neues Mitglied auf der Jahrestagung begrüßen zu können.

Last but not least: Begrüße ich **Chris Jakeman** in unserer Runde. Chris ist auf der Jahrestagung in Dingden der Forth Gesellschaft e.V. beigetreten. Vielen ist er bekannt durch sein höchst aktives Wirken in der FIG UK, als Redakteur und emsiger Autor der Forthwrite, als Initiator und Koordinator verschiedener Hard- und Software-Projekte unserer englischen Forth-Freunde und durch das internationale Web-Forth-Projekt, an dessen Entwicklung er großen Anteil hat und das er seit einiger Zeit leitet. Wir haben auf der Tagung seinen Vortrag über dieses Projekt und über die englische Forth Interest Group im allgemeinen gehört und konnten viele interessante Unterhaltungen mit ihm führen. Chris hat MAF (Minimal Ansi Forth) zu didaktischen Zwecken geschrieben und ist durch Veröffentlichungen nicht nur in der Forthwrite, sondern auch in der VD bekannt und geachtet.

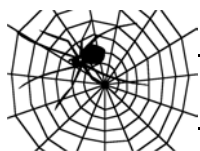
*Chris, please accept our warmest welcome as a new member of Forth Gesellschaft e.V., new to Forth Gesellschaft but not in the least new to Forth! We are honoured and proud to have you with us.*

---

**Forthgesellschaft goes Lego:** Auf der Jahrestagung in Dingden wurde beschlossen, dass die Forthgesellschaft zwei Lego-RCX Elemente und dazugehörig IR-Sendetürme erwirbt und diese Michael Kalus und Adolf Krüger zur Verfügung stellt.

Dies ist inzwischen geschehen. Beide haben mit ihrer Arbeit begonnen und erste Kontakt zu den RCXsen aufgenommen. Ziel ihrer Bemühungen ist zum einen die genaue Analyse des IR-Sendeprotokolls zwischen dem RCX und dem Sendeturm/PC-Gespann. Zum anderen das Erforschen möglicher Verbesserungen in dieser Kommunikation. Alles hängt davon ab, wie 'flexibel' die Hardware der RCX-Steine aufgebaut ist Zur Erinnerung, viele sendende RCXse in einem Raum stören sich gegenseitig!  
Michael, Adolf viele drücken Euch herzlich (die Daumen ;-)

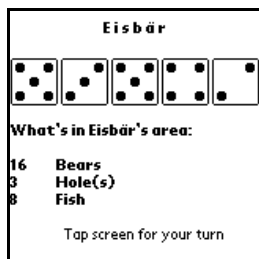




Unter der Webadresse

<http://home.t-online.de/home/erwin.schomburg/indexg.htm> hat

Erwin Schomburg eine Homepage kreiert, mit der er sich zum Ziel gesetzt hat, Lehrbeispiele zu **QuartusForth** auf dem Palm PDA zur Verfügung zu stellen. Es gibt ein kleines How To und einige 'verspielte' (grafische) Applikationen. Fast tut es mir leid, dass ich keinen Palm habe. MB



Harald Schreiber zeigt unter

<http://home.t-online.de/home/harald.schreiber/rechner.htm> ein

Java-Applet, das einen UPN-Rechner darstellt. Leider? Ist es nur von Forth inspiriert. Harald Schreiber hat dazu seine eigene Makro-Sprache **VEEMAL** (Very Easy Expandable MAkro Language) entwickelt. Immerhin, viele Makros tragen Forthnamen, man kann sich also leicht eingewöhnen.

Es gibt eine sehr experimentelle Portierung von **pForth** durch Marcus Gröber für die Nokia Handyfamilie 9000. Die aktuelle Version steht unter

<http://www.yaws.dk/communicator/software/pforth/home.html>

zum Downloaden bereit. Wer also schon immer mit seinem 'Telefon' mehr als nur reden wollte – jetzt geht's!

**MINOTAUR** ist der Versuch in Tcl verschiedene Programmiersprachen 'embedded' zu betreiben. Welche Sprache dabei der Aufrufer, und welche die aufgerufene Sprache ist, ist nicht immer leicht zu entscheiden. Jedenfalls kann man so das Paar Tcl/pForth betreiben. Ob eventuell auch Tcl/Tk/pForth möglich ist? Ich weiß es nicht, aber vielleicht kann das ja ein Vereinsmitglied herausfinden? Nachsehen kann man bei <http://www.equi4.com/minotaur/minotaur.html>

### Forth in Taiwan, Amerika und England - Ting, eForth, WebForth und anderes

Die Website der FIG-Taiwan ist jetzt unter [www.figTaiwan.org](http://www.figTaiwan.org) zu finden. Sam Suan Chen hat sich wieder gemeldet und unseren Webmaster und Direktor, Dr. Ulrich Hoffmann (Ulli, wir gratulieren Dir zur Wahl in diese Ämter!), gebeten, auf unserer Homepage einen Link nach Taiwan einzurichten. Wir sind mit Taiwan "verlinkt".

Sam geht in seiner E-Mail an mich, Fred Behringer, unter anderem auf Professor C.H. Ting von der FIG US ein. "Ting" (wie nach dessen eigenen Worten seine Freunde ihn nennen), Amerikaner chinesischer Abstammung, war wesentlich an der Gründung der FIG-Taiwan beteiligt. C.H. Ting, das wissen wir von Henry Vinerts' Reports aus Amerika her, hält sich regelmäßig längere Zeit in Taiwan auf, hält Vorträge an der Universität, führt Kurse durch und ist auch sonst sehr rührig. C.H. Ting hat, so Sam, gerade ein Buch mit dem Titel "Die eForth-Bibel" (in chinesischer Sprache) fertiggestellt und in Taiwan herausgebracht. In diesem Buch, so Sam, wird Tings Lebenswerk rund um Forth zusammengestellt.

Nun, etwas Ähnliches hat C.H. Ting, wir berichteten schon darüber, mit seiner kürzlich herausgegebenen CD (diesmal nicht auf Chinesisch) getan. Ting hat mir eine solche CD für 30 Dollar (5 Dollar Porto) geschickt und in seinem Begleitschreiben hat er seine besten Grüße an die Forth-Freunde in

Deutschland bestellt. Die CD enthält die reichlich gefüllten Unterverzeichnisse \eforth1, \eforth2, \f83, \figForth, \P-Series und \win32for.

C.H. Ting hat, so Sam, auch die eForth Company in Taiwan gegründet, die sich vorwiegend mit der Entwicklung eines eForth-Chips beschäftigt, der speziell auf die Verarbeitung der etwa dreißigtausend chinesischen Schriftzeichen ausgerichtet ist. Er soll die TrueType-Technik beherrschen können.

Von eForth habe ich, haben wir, das erste Mal gehört, als Michael Losh (FIG US) sein auf Java gegründetes Forth herausbrachte. Mike nannte es zunächst jeForth (Java-eForth). Mike hat sich dann mit der FIG UK zusammengetan und sein Tun an einem internationalen Projekt auf die Errichtung einer interaktiven Website mit Tutorials zur Förderung des Nachwuchses ausgerichtet.

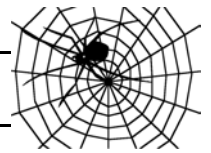
Aus jeForth wurde JansForth, nachdem klar wurde, daß eine Ansifizierung nicht schaden könne, wenn man Akzeptanz finden möchte. Mike Losh übergab schließlich die Koordination der internationalen Bemühungen an Chris Jakeman, den Redakteur der englischen Forthwrite, den wir ja auf der Forth-Tagung 2001 als Special Guest begrüßen durften. Mike trat ab dann als Berater in den Hintergrund, das Projekt wurde in "WebForth" umgetauft und das internationale Team, dem drei Engländer, ein in Japan unterrichtender Engländer, ein Amerikaner, ein Spanier und ein Deutscher angehören (neuerdings auch ein Chinese), nennt sich seither "WebForth-Team". Der Deutsche bin ich und Sam Suan Chen ist der Chinese. Sam spielt mit dem Gedanken, die WebForth-Site auch ins Chinesische zu übertragen. (Es geht um ein Java-Applet als Forth-System, das in ein Dickicht von HTML-Seiten eingebettet ist, welche wiederum per Mausclick in die englische, deutsche, spanische und wohl bald auch chinesische Richtung gelenkt werden können. Die HTML-Seiten stellen Lehrstücke und Programmierbeispiele in WebForth bereit, mit denen sich der Web-Besucher interaktiv einarbeiten kann.)

beh.

**Raphael Deliano** der Verfasser und Herausgeber der feinen Insiderzeitschrift **embedded** teilt mit, dass er die **embedded** ab sofort als PDF Datei zum Download anbietet. Gleichzeitig wird das Erscheinen der Papierversion eingestellt. Auf begründeten Wunsch kann in Einzelfällen weiterhin eine Papierausgabe versandt werden.

Online gibt es nun einige zusätzliche Serviceleistungen: bisherige Versionen der **embedded** stehen nun ebenso wie Listings zum Download bereit, man kann sich in eine Benachrichtigungsliste eintragen lassen. Die **embedded** liegt damit in der sechsten Ausgabe vor. <http://www.embeddedFORTH.de>

Chuck Moore hat nun eine eigene Homepage. Dort gibt's einen Link zu <http://www.colorforth.com> (meine Sprachkenntnisse reichen nicht aus, zu entscheiden, ob die damit verbundene Danksagung ironisch gemeint ist). Aber! Alle wirklich wesentlichen und die neuesten Informationen dazu gibt es eben bei Chucks persönlicher <http://www.mindspring.com/~chipchuck/> Weiterhin finden sich dort Links zu **VLSI-Tools** und ein Chip **25x**. Ein Array von 25  $\mu$ C, natürlich programmierbar mit ColorForth.

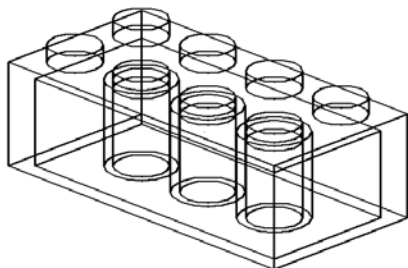


MPE reagiert auf die angespannte Marktsituation mit einem deutlichen Preisnachlass auf VFX Forth für Windows (95, 98, NT, 2000, ME). Interessanter ist aber die Möglichkeit eine kostenlose Trialversion für 30 Tage 'downzuloaden'. Einzige Einschränkungen: Es können keine stand-alone-Applikationen hergestellt werden und bei jedem Start meldet sich ein 'nerviger' Registrierungshinweis.

<http://www.mpeltd.demon.co.uk/arena.htm#trial>

Ralph Hempels pbForth hat im Moment noch die Versionsnummer 2.12, aber er arbeitet an einem 'redesign'. Mit einigen neuen Features: So soll der Kernel noch mehr Multitaskingfähigkeiten erhalten. Insbesondere sollen die meisten I/O Operationen (Motoren, Sensoren, Buttons, Display ...) ihre eigenen vordefinierten Tasks erhalten. Möglichkeiten für weitere 'Helper' -Tasks sind vorgesehen. Nun ist ein Ausschalten über den Powerknopf möglich, das lästige Batterieentfernen bzw. Stromlosmachen entfällt damit. Das 'neue' pbForth wird sich an Chris Jakemans (seit April Mitglied der Forth Gesellschaft e.V.) MAF/PAF System anlehnen. (MAF = Minimals Ansi Forth) Der neue Kernel läuft bereits im High-Level zur Geschwindigkeitsoptimierung müssen noch einige Worte in Maschinendprache (um)geschrieben werden. Der kommende pbForthkernel wird kleiner als 9 Kb sein, Ralph rechnet damit, dass er Ende September fertig ist. Ich warte schon gespannt auf das 'neue' pbForth.

<http://www.hempeldesigngroup.com/lego/pbFORTH>



**Jeremy Fowell** hat den Vorsitz bei der FIG UK übernommen hat. Er löst damit Chris Hainsworth ab. Jeremy Fowell war bisher Mitglied der FIGUK-Führungsgruppe und Initiator des gerade laufenden Hardware-Projekts, das bei den FIGUK-Mitgliedern starken Anklang gefunden hat.

Congratulations!

#### In eigener Sache:

Das Internet ist riesig und leider schwach strukturiert. Viele interessante Informationen und Neuigkeiten aus der Forth-Welt bleiben verborgen und erreichen potenzielle Adressaten nicht. Helfen Sie mit, die vielen kleinen Informationssplitter zu sammeln und zusammenzutragen. Senden Sie interessante Web-Adressen und Mitteilungen an die Redaktionsadresse. Ein wahrer Glückstreffer wäre es, erklärte sich jemand bereit, die betreffende Rubrik in der Vierten Dimension zu pflegen. Denn: Die Arbeit an der Vierten Dimension macht Spaß – noch mehr Spaß macht sie (so wie jede echte Freude) wenn sie geteilt wird.

## Leserbrief

Hallo Martin,

Im Zusammenhang mit Besprechungsarbeiten habe ich mir den CAM-Artikel ("umgekehrte" Speicher) von Ulrich Paul (VD 2/2001) noch einmal genauer angesehen. Der Autor kann schreiben! CAMs mit Hundhammer-Deutsch (Lübke-Englisch: Equal goes it loose) gewürzt - nicht schlecht! Man sollte ihn zum öfteren Schreiben animieren!

Assoziative Speicher erinnern mich irgendwie an hydraulische Simulationsmodelle für Probleme der Linearen Optimierung: Kommunizierende Röhren (mit Öffnungventilen auch nach unten). Unten "drückt man drauf" ("schiebt" die Zielfunktion "nach oben"), oben kommen an den einzelnen Zweigen unter Beachtung von Nebenbedingungsschranken die für eine optimale Lösung zuständigen Variablenwerte in Form von Flüssigkeitsständen heraus. Zugegeben, das sind bei den heutigen schnellen, zuverlässigen und genauen Digitalmaschinen überholte Vorstellungen. Aber CAMs für die digitale Simulation ...

Mein Interesse ist geweckt. Wenn ich im Elektor oder in der Vierten Dimension eine PC-Einsteckkarte mit CAMs zum Nachbauen finde, werde ich mich draufstürzen.\*

Auch Ulrich Pauls Leserbrief (ebenfalls VD 2/2001) über das immer wieder zum Nachdenken anregende Thema "Forth gut - Forth nicht gut" hat mir, vor allem in den Formulierungen, gefallen.

Gruß

Fred (Behringer)

Ganz knapp vor Redaktionsschluss erlangte Fred Behringer folgende Information: Chris Jakeman berichtet in der Forthwrite 112 von der Jahrestagung der Forth Gesellschaft. Er ist angenehm überrascht und wünscht sich eine ähnliche Veranstaltung: „Warum haben wir für unsere FIG UK keine Tagungen von ähnlichem Ausmaß? Die Mitgliederzahl liegt etwa in gleicher Höhe und unsere Anreiseentfernungen sind sogar kürzer. Vorausgesetzt, wir finden einen energiegeladenen Freiwilligen, könnten wir wirklich ernsthaft darüber nachdenken. Für das nächste Jahr hätte ich jedoch einen anderen Vorschlag:

Die Forth-Gesellschaft hat lange Jahre Erfahrung in der Veranstaltung solcher Tagungen, und wenn wir nach dem langen Winter einmal für ein Wochenende von unserer Insel wegkommen könnten, wäre das auch nicht schlecht. Warum wollen wir uns nicht einfach zusammentun und auf der nächsten Forth-Tagung in Deutschland (München, 19.-21. April 2002) mit einer kleinen englischen Abordnung erscheinen? Ein herzlicher Empfang ist uns garantiert sicher.“ Nun ich denke viele Deutsche Forthler freuen sich schon jetzt darauf, britische 'Kollegen' kennenzulernen. Den Termin kann man sich auf jeden Fall vormerken: nächste Tagung in München vom 19. - 21 April  
Organisator H. Schnitter

\* Wer macht's?



## Lego-Roboter und arithmetisierte Logik in Forth

Fred Behringer <behringe@mathematik.tu-muenchen.de>

**Stichworte:** Robotersensoren, 16-Bit-Forth, boolesche Funktionen von bis zu 16 Variablen, Bedingungssteil für IF-Strukturen, Disjunktive Normalform, definierendes Forth-Wort DNF, Arithmetisierung mit 1 oder -1 als "wahr", Multilinearform, definierendes Forth-Wort MLF, Eingaben in Tabellenform im Quelltext, Forth-Wort DATA zur Einleitung der Tabelle unter Festhaltung des Stackstandes (SP), Forth-Wort BNS (bitwise not smaller than) als zentraler Operator, systematische Umwandlung einer DNF in die MLF.

Martin Bitter hat mit seinen hervorragenden Vorträgen und der praktischen Arbeit im Media-Night-Zelt der Forth-Gesellschaft im Juli 2000 in Vaterstetten das Zeitalter des Schulprojekts "Roboterprogrammieren in Forth" eingeleitet und ich stecke jetzt ehrenamtlich mitten im Kurs-Geschehen. "Programmieren in Forth ist kinderleicht", war mein Spruch den Schulverantwortlichen gegenüber. "Können die Kinder denn nun schon programmieren?", wurde ich neulich gefragt. Nein, das können sie natürlich noch nicht. Ganz so leicht ist es denn nun auch wieder nicht.

Wichtig an einem Roboter sind die Sensoren. Der RCX-Baustein hat nur drei Sensoreingänge und die Lego-Sensoren sind teuer. Man kann sich aber leicht Sensoren selber bauen. Ich habe mir inzwischen 16 Sensoren zusammengebastelt. An den Sensor-Eingängen des RCX liegen A/D-Wandler. Also kann man an einem einzigen Eingang mehrere Sensoren über ein Widerstandsnetzwerk hinzu- oder abschalten. Die einfachsten Möglichkeiten sind Parallel- oder Reihenschaltung der Widerstände. Es ergibt sich ein nicht ganz einfaches mathematisches Problem der optimalen Verteilung der den einzelnen Sensorkombinationen (gedrückt oder nicht gedrückt) zugeordneten Widerstandswerte. Ich werde mich darüber an anderer Stelle auslassen. Damit man weiß, worüber wir reden, nur soviel: Die DNF einer jeden boolesche Funktion mit 16 Variablen (Sensoren) hat 65536 Vollkonjunktionen. Jede von diesen tritt in bestimmten Funktionen auf, in anderen nicht. Es gibt also  $2^{65536}$  verschiedene solcher Funktionen. (Die mutmaßliche Größenordnung der Zahl der Elektronen im Weltall liegt bei  $2^{230}$  [6]. Wieviele Weltalls wären nötig, um  $2^{65536}$  Elektronen zu umspannen? Jeder Kommentar erübrigt sich - und 16 Lego-Sensoren ist wirklich keine große Zahl !)

Man stelle sich einen Roboter vor, der vorn und hinten je zwei Berührungssensoren hat, je einen rechts und einen links, und außerdem noch zwei Lichtsensoren zur differenziellen Verfolgung einer Linie auf dem Boden. Die Sensoren liefern die logischen Werte 0 und 1 und das Zusammenwirken soll auch durch 0 und 1 beschrieben werden können. Mathematisch gesehen, ist das eine boolesche Funktion von 6 Variablen. - Ich werde es im weiteren Verlauf mit Bits in Bytes und 16-Bit-Forthsystemen zu tun haben und richte meine Betrachtungen gleich auf boolesche Funktionen von 16 Variablen (die durch Bits in einem 16-Bit-Wort dargestellt werden) aus.

Vor einiger Zeit habe ich in der Vierten Dimension [2] über Disjunktive Normalformen (DNF) im Zusammenhang mit Conways "Game of Life" geschrieben und Forth-Vorschläge

gemacht. Eine DNF ist eine disjunktive Verknüpfung von Vollkonjunktionen. Eine Vollkonjunktion ist eine konjunktive Verknüpfung sämtlicher Variablen, wobei die einzelnen Variablen entweder in affirmativer oder in negierter Form auftreten. Näheres dazu findet man beispielsweise in [5]. Es folgt eine einfache DNF zweier Variablen (in Forth-Notation):

**X1 X2 AND X1 X2 NOT AND OR**

Vorsicht: Die vielgeübte Praxis, alle Werte ungleich 0 als "wahr" zu betrachten, ist hier im allgemeinen unzulässig und X1 und X2 sind als VALUES im ANS-Sinn aufzufassen, in F83 mit dem zugehörigen TO zum Wertebeladen beispielsweise wie folgt nachimplementierbar:

```
: VALUE CREATE , DOES> @ ;
: TO ' 2+ ! ;
```

Interpretiert man hier die Wahrheitswerte als ganze Zahlen, betrachtet 0 als "falsch" und 1 als "wahr", setzt für die Konjunktion die Multiplikation, für die Disjunktion die Addition und für die logische Negation von X2 den arithmetisierten Ausdruck **1 X2 -**, dann erhält man:

**X1 X2 \* X1 1 X2 - \* +**

Die Gleichwertigkeit der beiden Ausdrücke läßt sich leicht einsehen: In einer DNF kann immer nur jeweils eine einzige Vollkonjunktion den Wert 1 annehmen, womit die Disjunktion auf die Addition hinausläuft. - Das ist der erste Schritt einer "Arithmetisierung der Logik". (Natürlich läuft das Ganze in diesem Beispiel auf X1 hinaus.)

Neben der DNF gibt es noch eine andere Normalform, eine, die ganz stark von der Arithmetisierung abhängt: die Multilinearform (MLF). Ich möchte zeigen, daß in gewissen Fällen die MLF leichter zu handhaben ist als die DNF.

**Arithmetisierung:** Man kann die Aussagenlogik (Boolesche Algebra, Schaltalgebra) bekanntlich allein über Konjunktion und Negation aufbauen. Interpretiert man die Wahrheitswerte (0 für falsch und 1 für wahr) als ganze Zahlen mit der üblichen Arithmetik, dann bekommt man für die Multiplikation dieselbe Wertetafel wie für die Konjunktion. Die Disjunktion läßt sich über das De-Morgan-Theorem über Konjunktion und Negation ausdrücken. Setzt man für die Negation der Variablen x den Ausdruck **1-x**, dann erhält man (in Infixnotation!):

**¬x = 1-x    x∧y = x\*y**

und daraus (De Morgan)

**x∨y = x+y-x\*y    {0,1} wahr = 1**

Diese Ausdrücke erinnern stark an die Kolmogoroffschen Axiome in der Wahrscheinlichkeitstheorie. Stellt man sich darunter Verknüpfungen von Zählern (Geiger-Müller-Rohre, Szintillationszähler) von Zufallsereignissen (Höhenstrahlungspartikel, durch diese erzeugte Spannungsimpulse) vor und interpretiert man Wahrscheinlichkeit als "Grenzwert von relativen Häufigkeiten", dann kann man **x∧y** mit dem Durchschnitt zweier Ereignismengen, **¬x** mit der Komplementärereignismenge und **x∨y** mit der Vereinigung zweier Ereignismengen in Verbindung bringen. **x∨y** entspricht dann also nicht ganz einfach nur der Summe der relativen Häufigkeiten der Einzelelementarereignisse, **x+y**, sondern





man muß davon noch die relativen Häufigkeiten derjenigen Elementarereignisse je einmal abziehen, die als Paar, also gleichzeitig, aufgetreten sind, und die sind aber genau in  $x\Delta y$ , also in  $x*y$ , vertreten. Solche Impulszählfragen traten früher bei Koinzidenzmessungen (Winkelabhängigkeit der Höhenstrahlung) auf, als man  $x\Delta y$  über  $x\vee y$  bestimmte, was wiederum durch einen in den Katodenweg einer Doppeltriode gelegten Widerstand leicht bewerkstelligt werden konnte. - Heute kauft man man sich einfach ein passendes IC.

In Forth arbeiten wir gern mit **-1** als "wahr" (und **0** als falsch). Das geht auch:

$$\neg x = -(1+x) \quad x\Delta y = -(x*y)$$

und daraus (De Morgan)

$$x\vee y = x+y+x*y \quad \{0, -1\} \text{ wahr} = -1$$

**Bezeichnung:** Ich lasse in der vorliegenden Arbeit bis zu 16 Variablen zu, die ich bitweise zu einem "Wort", einem 16-Bit-Feld (einer "Zelle" über zwei Bytes) vektoriell zusammenfasse. Variablen ohne Index (Vektorvariablen) kennzeichnen eine solche vektorielle Zusammenfassung (ein 16-Tupel), Variablen mit Index sind die in den einzelnen Bits aufbewahrten eigentlichen Variablen:  $x := (x_{15}, \dots, x_0)$ . Die Indizes werden in der üblichen Reihenfolge der Bitpositionen im Computerwort von rechts her hingeschrieben.

**Multilinearform:** Jede boolesche Funktion von n Variablen läßt sich eindeutig als Multilinearform (MLF) darstellen. Die Bestandteile einer MLF sind Produkte von affirmativen Variablen. Nicht alle Variablen brauchen vorzukommen. Negierte Variablen treten nicht auf. Die MLF-Bestandteile sind mit ganzzahligen Koeffizienten versehen, die auch negativ sein können. Genau wie bei einer DNF können bis zu  $2^n$  solcher Bestandteile vorkommen. (Schnelle Begründung: Es gibt  $2^n$  n-stellige Binärzahlen, also Verteilungen von Nullen und Einsen über n Bitstellen.) Für die allbekannte "Brückenschaltung" sieht die MLF (in Infix-Notation) wie folgt aus (das Multiplikationszeichen zwischen den Variablen wird hier und im weiteren Verlauf weggelassen):

$$\begin{aligned} & x_1x_0 + x_3x_2 + x_4x_3x_0 + x_4x_2x_1 - \\ & - x_3x_2x_1x_0 - x_4x_2x_1x_0 - x_4x_3x_1x_0 - \\ & - x_4x_3x_2x_0 - x_4x_3x_2x_1 + 2*x_4x_3x_2x_1x_0 \end{aligned}$$

Diese MLF hat 10 additive Bestandteile! Die DNF hätte 16 Vollkonjunktionen! Die MLF liegt also bei dem Verfahren, das ich gleich vorschlagen werde, besser als die DNF. Als Koeffizient der einzelnen Bestandteile tritt bei der vorliegenden MLF viermal **+1**, fünfmal **-1** und einmal **+2** auf.

Genauer gesagt, können in einer MLF bis zu  $2^n-1$  Bestandteile und dazu dann noch eine additive Konstante (mit dem Wert 0 oder 1) auftreten. Die additive Konstante hat den Wert 1, tritt also wirklich auf, wenn die betrachtete boolesche Funktion bei Eingangsbelegung  $(0, \dots, 0)$  den Wert 1 annimmt. Ich werde im folgenden unter "MLF-Bestandteil" immer auch diese gegebenenfalls (mit dem Wert 1 wirklich) auftretende additive Konstante in die Betrachtung mit einbeziehen.

### Verfahren (zur Ermittlung des Funktionswertes bei gegebener Variablenbelegung aus der MLF heraus):

In einer DNF wird zur binärzahligen Charakterisierung einer Vollkonjunktion jeder affirmativen Variablen eine **1** zugeordnet und jeder negierten Variablen eine **0**. In den MLF-Bestandteilen dagegen gibt es keine negierten Variablen. Ich indiziere die MLF-Bestandteile binär durch und ordne dabei jeder **auf tretenden** Variablen eine **1** zu und jeder **nicht auftretenden** eine **0**. Der weiter oben vorgekommene (multiplikative) Ausdruck  $x_4x_3x_2x_0$  erhält dementsprechend den (als Binärzahl aufzufassenden) Index **11101** (die Einbitvariable  $x_1$  kommt im Produkt nicht vor, daher die **0**) zugeordnet. Diese Art der Schematisierung wirkt sich insbesondere auf die Koeffizienten aus: Der dem Ausdruck  $x_4x_3x_2x_0$  zugeordnete Koeffizient soll mit  $a_{11101}$  bezeichnet werden. Die Indizierung der Variablen lasse ich von rechts her bei **0** beginnen, da diese, also die Vektorkomponenten, durch Bits im Doppelbyte repräsentiert werden und auch die Bitzählung im Computerwort üblicherweise bei **0** beginnt.

### Beim Compilieren:

(1) Erzeuge zwei Felder (**B** und **K**), deren Länge in Zellen der Anzahl der Bestandteile der MLF der zu bearbeitenden logischen Funktion entspricht. Als "Bestandteil" im eigentlichen Sinne sollen nur solche gelten, deren Koeffizient ungleich **0** ist.  $0*0*...*0$ , also "keine Variable", muß gegebenenfalls auch als "Bestandteil" gezählt werden, nämlich dann, wenn die Konstante  $a_0$  der betreffenden MLF den Wert **1** hat, wenn also die zugehörige boolesche Funktion bei Variablenbelegung  $(0, \dots, 0)$  den Wert **1** hat.

(2) Speichere in **B** die Indizes der Koeffizienten der MLF-Bestandteile ab und in **K** die Koeffizientenwerte.

### Zur Laufzeit:

(3) Lege den Anfangswert **0** auf den Stack.

(4) Bei gegebener Variablenbelegung  $x$  (als Binärzahl-Eingabevektor im obigen Sinne) gehe in einer Schleife alle Indizes  $b$  aus **B** durch und bilde  $x \ b \ \text{AND}$ . Es werden alle Komponenten von  $x$ , die im betreffenden Index  $b$  den Wert **0** haben, ausgeblendet. Sie "zählen nicht". Im selben Schleifenschritt prüfe gleich anschließend  $x \ b \ \text{AND}$  auf Gleichheit mit  $b$ , d.h., untersuche  $x \ b \ \text{AND} \ b = .$  Liegt Gleichheit vor, dann addiere den zugehörigen Koeffizienten  $k$  aus **K** zum obersten Stackwert hinzu (Vorzeichen beachten).

(5) Nach Beendigung der Schleife liegt auf dem Stack der Wahrheitswert (**0** = "falsch", **1** = "wahr") der untersuchten logischen Funktion bei der betreffenden Belegung  $x$  als Binärzahl-Eingabevektor.

**Erklärung zu (4):** Ist in der gerade untersuchten Belegung  $x$  auch nur eine einzige Komponente **0**, während der Index  $b$  des momentanen Schleifenschritts in dieser Komponente den Wert **1** hat, dann zieht diese  $x$ -Komponente den zugehörigen MLF-Bestandteil (als Faktor in einer Mehrfachmultiplikation) auf **0**



## ... arithmetisierte Logik

und dessen Koeffizient  $a_b$  wird nicht in den Summierprozeß auf dem Stack mit aufgenommen. Mit anderen Worten, **b** liefert unter diesen Umständen bei Belegung **x** keinen Summenanteil zur Berechnung des logischen Wertes aus der MLF heraus.

Es folgt das Programmlisting (in TurboForth oder ZF). Hier und weiter unten verwende ich folgende Operation:

```
: BNS ( a b -- fl )
  TUCK AND = ;
  \ a bitwise not
  \ smaller than b
```

Wenn **b** an mindestens einer Bitstelle eine **1** hat, die in **a** nicht vorkommt, dann ist **fl=0**. Sonst ist **fl=-1** (das Minuszeichen ist ein Zugeständnis an Turbo-Forth und andere Forth-Systeme, in denen **IF**-Konstruktionen, theoretisch jedenfalls, -1 als "wahr" erwarten). **BNS** verkörpert die übliche Größer/gleich-Relation auf der Menge aller booleschen Vektoren. Das ist eine Halbordnungrelation (reflexiv, transitiv und antisymmetrisch). Halbordnung heißt, es können auch Elemente **a** und **b** auftreten, die miteinander nicht vergleichbar sind, bei denen also weder **a** bitweise nicht kleiner als **b** noch **b** bitweise nicht kleiner als **a** ist.

```
: DATA ( -- sp )
  -1 [ HERE ] LITERAL 2- SP@ 2- SWAP ! ;
```

Mit **DATA** habe ich mir ein (hoffentlich) leicht zu merkendes Wort konstruiert, das die Datenliste (Koeffizient und Index paarweise zusammengeführt) einleitet und für den anschließenden Aufbau (über das definierende Wort **MLF**) eines "Objektes" (Algorithmus und zugehöriger Datenkörper) die Stackposition aufbewahrt und bei nochmaligem Aufruf (im definierenden Wort) wieder zur Verfügung stellt. Seine Funktion ist gleichbedeutend mit:

```
-1 CONSTANT XXX und dann
jedesmal
XXX SP@ 4 - TO XXX
```

Bei dem gleich folgenden definierenden Forth-Wort **MLF** (obenstehender Kasten) werden die weiter oben mit **B** und **K** bezeichneten Felder reißverschlussartig ineinander geschoben, immer je ein Element (eine "Zelle") aus **B** (Index) und eins aus **K** (Koeffizient).

Daß die Brückenfunktion bei der MLF besser liegt als bei der DNF, ist nur die halbe Wahrheit. Ich werde im folgenden ein dem obigen Forth-Wort **MLF** ähnliches definierendes Forth-Wort, **DNF**, aufbauen.

```
: MLF ( <name> Compilezeit: data a1 k1 .. an kn -- )
  ( Laufzeit: belegung -- fl )
  CREATE HERE >R 0 , \ Freilassen fuer Laenge
  DATA \ Stackstand zu Anfang der Daten
  BEGIN
  DUP SP@ <> \ Wenn nicht Anfang,
  WHILE
  -ROT , , \ dann die naechsten 2 Elemente aufnehmen
  REPEAT
  2DROP \ Anfangsstackstand zweimal beseitigen
  R> HERE OVER -
  SWAP ! \ Laenge eintragen
  DOES> 0 SWAP \ Vorgabe zum Koeffizientensammeln
  DUP @ OVER + SWAP 2+ \ "BOUNDS" + 2 zur unteren Grenze
  DO
  OVER I 2+ @ BNS \ MLF-Bestandteil = 1 ?
  IF I @ + THEN \ Koeffizient hinzuaddieren
  4 +LOOP \ 2 Zellen weiter
  NIP NEGATE ; \ Wert bei dieser Belegung = TRUE ?

\ == Anwendungsbeispiel 1 (2-aus-3-Funktion, siehe unten) ==
2 BASE ! DATA 011 1 101 1 110 1 111 -10 MLF 2AUS3?

\ == Anwendungsbeispiel 2 (Brueckenfunktion) ==
DATA 00011 1 01100 1 11001 1 10110 1
01111 -1 10111 -1 11011 -1 11101 -1
11110 -1 11111 10 MLF BRUECKE? HEX
```

Anstelle der MLF-Bestandteile treten darin die Vollkonjunktionen der DNF auf. Da als (nichtverschwindender) Koeffizient einer Vollkonjunktion nur die 1 in Frage kommt, kann man sich die Hälfte des Speicherplatzes sparen. Man braucht (beim Compilieren) nur die Indizes der in der DNF tatsächlich auftretenden Vollkonjunktionen (z.B. **1001** für **x3(1-x2)(1-x1)x0**) hintereinanderweg abzuspeichern und später (während der Laufzeit) der Reihe nach abzufragen. Es braucht keine Koeffizientensumme gebildet zu werden, da bei Belegung **x** nur höchstens eine einzige Vollkonjunktion den Wert 1 annehmen kann. Diese bestimmt schon den Wert der gesamten booleschen

```
: DNF ( <name> Compilezeit: data v1 .. vn -- )
  ( Laufzeit: belegung -- fl )
  CREATE HERE >R 0 , \ Freilassen fuer Laenge
  DATA \ Stackstand zu Anfang der Daten (Vollkon-
  BEGIN \ junktions)
  DUP SP@ <> \ Wenn nicht Anfang,
  WHILE
  SWAP , \ dann naechste Vollkonjunktion aufnehmen
  REPEAT
  2DROP \ Anfangsstackstand zweimal beseitigen
  R> HERE OVER -
  SWAP ! \ Laenge eintragen
  DOES>
  FALSE -ROT \ Vorgabe, falls kein fruehzeitiger Ausssprung
  DUP @ OVER + SWAP 2+ \ "BOUNDS" + 2 zur unteren Grenze
  DO
  DUP I @ = \ Vollkonjunktion = Belegung ?
  IF
  2DROP -1. LEAVE \ Belegung und FALSE weg und Ausssprung
  THEN
  2 +LOOP \ 1 Zelle weiter
  DROP ; \ Ueberfluessigen Teil von -1. beseitigen

\ == Beispiel 1 (2-aus-3-Funktion) ==
2 BASE !
DATA 011 101 110 111 DNF 2AUS3?

\ == Beispiel 2 (Brueckenfunktion) ==
DATA 00011 00111 01011 01100 01101 01110 01111 10110
10011 10111 11011 11100 11101 11110 11111 11001 DNF BRUECKE? HEX
```



Funktion bei Belegung  $x$ . Bei  $m$  Vollkonjunktionen benötigt der Suchprozeß im Mittel nur  $m/2$  Schritte. Sobald eine Vollkonjunktion bei gegebener Variablenbelegung den Wert 1 liefert, ist die Suche beendet. Hinzu kommt der Vorteil, daß man die Vollkonjunktionen einer DNF sofort angeben (aufschreiben, eintippen oder im Quelltext festhalten) kann. Man braucht ja "nur" die Wertetafel der betrachteten booleschen Funktion durchzugehen. Es folgt das definierende Wort **DNF** zur Konstruktion eines in Kontrollstrukturen verwendbaren Bedingungswortes (eines "Conditionals") zu einer (als Tabelle) beliebig vorgebbaren booleschen Funktion von bis zu 16 Variablen (siehe Kasten).

Die DNF einer ins Auge gefaßten booleschen Funktion zu gewinnen, ist verhältnismäßig leicht: Man geht sämtliche Variablenbelegungen durch und sieht nach, ob die "zugehörige" Vollkonjunktion ( $\neg x$  für Variablenbelegung **0**,  $x$  für Variablenbelegung **1**) den Wert **1** liefert oder **0**. Bei **1** wird sie in die DNF aufgenommen, bei **0** nicht. Die Koeffizienten der MLF zu ermitteln, ist nicht ganz so einfach. Zur Erklärung des gleich zu besprechenden systematischen Verfahrens gehe ich von der 2-aus-3-Funktion aus. (Beispiel aus der Zuverlässigkeitstheorie [1]: Ein dreimotoriges Flugzeug bleibt intakt, wenn mindestens 2 von 3 Komponenten (Motoren) intakt sind).  $x$  ist die vektorielle Zusammenfassung der einzelnen Einbitvariablen,  $x := (x_2, x_1, x_0)$ :

Funktionstafel der 2-aus-3-Funktion

$x_2$	$x_1$	$x_0$	$f(x)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Die Koeffizienten gewinnt man hieraus aus:

$$\begin{aligned}
 f(0,0,0) &= 0 = a_{000} \\
 f(0,0,1) &= 0 = a_{001} + a_{000} \\
 f(0,1,0) &= 0 = a_{010} + a_{000} \\
 f(1,0,0) &= 0 = a_{100} + a_{000} \\
 f(0,1,1) &= 1 = a_{011} + a_{000} + a_{001} + a_{010} \\
 f(1,0,1) &= 1 = a_{101} + a_{000} + a_{001} + a_{100} \\
 f(1,1,0) &= 1 = a_{110} + a_{000} + a_{010} + a_{100} \\
 f(1,1,1) &= 1 = a_{111} + a_{000} + a_{001} + a_{010} + a_{100} + a_{011} + a_{101} + a_{110}
 \end{aligned}$$

$a_{011}$  bedeutet den Koeffizienten von  $x_1x_0$ ,  $a_{101}$  den von  $x_2x_0$  usw. Verkürzt dargestellt:

$$\begin{aligned}
 0 &= a_{000} & 0 &= a_{001} & 0 &= a_{010} & 0 &= a_{100} \\
 1 &= a_{011} & 1 &= a_{101} & 1 &= a_{110} \\
 1 &= a_{111} + 1 + 1 + 1
 \end{aligned}$$

also für die 2-aus-3-Funktion :

$$MLF(x) = x_1x_0 + x_2x_0 + x_2x_1 - 2x_2x_1x_0$$

$a_{000}$  gewinnt man sofort aus  $f(0,0,0)$ .

Mit  $a_{000}$  gewinnt man nacheinander  $a_{001}$ ,  $a_{010}$  und  $a_{100}$ .

Daraus gewinnt man dann nacheinander  $a_{011}$ ,  $a_{101}$  und  $a_{110}$ .

Und aus diesen Ergebnissen gewinnt man schließlich  $a_{111}$ .

Und so auch bei weiteren Variablen.

Wie man leicht einsieht, kann die MLF einer gegebenen booleschen Funktion  $f$  von  $n$  Variablen ganz allgemein wie folgt dargestellt werden:

$$f(x) = [a_b (x_1^{b_1} \dots x_n^{b_n})]$$

für jede Eingangsbelegung  $x=(x_1, \dots, x_n)$  aus  $B^n$

$\Sigma$  soll sich über alle  $b=(b_1, \dots, b_n)$  aus der Menge  $B^n$  der  $n$ -komponentigen booleschen Vektoren erstrecken, deren Komponenten (nur) die Werte **0** oder **1** annehmen können. Die Beziehung gilt für jedes  $x$  aus der eben genannten Menge  $B^n$  aller  $n$ -komponentigen Vektoren mit **0** oder **1** für die Komponenten. Die Funktion  $f$  wird durch die Koeffizienten  $a_b$ , von denen - und das hoffen wir in der praktischen Anwendung sehr - einige auch **0** sein können, eindeutig festgelegt. Dem mathematischen Sprachgebrauch folgend, habe ich in dieser Aussage die Indizierung von **1** bis  $n$  laufen lassen, und nicht von  $n-1$  bis **0**. Im Rest der Arbeit verwende ich dann wieder die in der Computertechnik übliche Bitindizierung von  $n-1$  bis **0**.

**Verfahren (am Fall  $n=3$  erläutert, aber allgemein aufzufassen):**

**(0)** Aus der Belegung  $x=(0,0,0)$  bekommt man die additive Konstante  $a_{000}$ .

**Begründung:** Jeder andere MLF-Bestandteil enthält bei der Belegung  $x=(0,0,0)$  mindestens eine Variable  $x_i$  (als Faktor), die den Wert  $x_i=0$  aufweist und daher in der Gesamtbetrachtung (bei eben dieser Belegung  $x=(0,0,0)$ ) nicht zum Tragen kommt.

**(1)** Untersuche der Reihe nach alle Belegungen, bei denen genau eine Variable den Wert **1** hat, also beispielsweise die Belegung  $x=(0,1,0)$ . Hieraus und aus dem bereits bekannten Wert für die additive Konstante, also im Beispiel  $a_{000}$ , läßt sich der betreffende Koeffizient, also beispielsweise  $a_{010}$  berechnen.

**Begründung:** Bei der Belegung  $x=(0,1,0)$  weist jeder andere MLF-Bestandteil mit einer (anderen als der eben besprochenen) Variablen oder mit sogar zwei und mehr Variablen mindestens eine Variable  $x_i$  mit dem Wert  $x_i=0$  auf, die den betreffenden MLF-Bestandteil auf **0** zieht und ihn folglich nicht zum Tragen kommen läßt.

**(2)** Untersuche der Reihe nach alle Belegungen, bei denen genau zwei Variablen den Wert **1** haben, also beispielsweise die Belegung  $x=(1,0,1)$ . Hieraus und aus den bereits bekannten Werten für die additive Konstante und für die Koeffizienten



der MLF-Bestandteile mit genau einer Variablen läßt sich der betreffende Koeffizient, also im eben genannten Beispiel  $a_{101}$ , berechnen.

**Begründung:** MLF-Bestandteile, die ein anderes Paar von Variablen ( $x_i, x_j$ ) oder sogar mehr als zwei Variablen enthalten, weisen mindestens eine Variable auf, die den Wert **0** hat und den betreffenden MLF-Bestandteil folglich auf **0** zieht und damit aus der Betrachtung herausfallen läßt.

**Achtung:** Bei den (schon ermittelten) Koeffizienten mit höchstens einer **1** im Index dürfen in der Summenbildung zur Ermittlung des booleschen Wertes der MLF nur solche berücksichtigt werden, deren Index bitweise kleiner/gleich dem Index des gerade betrachteten MLF-Bestandteils (hier mit zwei Variablen) ist. Ist auch nur ein einziges Bit im Index (des Koeffizienten mit höchstens einer **1** im Index) größer, so zieht die (dann ja tatsächlich auftretende) zugehörige Variable (als Faktor mit Wert **0**) den betreffenden MLF-Bestandteil auf **0**. Im allgemeinen Teil unseres obigen Beispiels darf zum Beispiel in Zeile 6, wo es um die Ermittlung von  $a_{101}$  geht,  $a_{010}$  nicht in die Summenbildung mit aufgenommen werden, da **010** nicht komponentenweise kleiner/gleich **101** ist. Faktisch kommt das in diesem Beispiel nicht zum Tragen, da der Koeffizient  $a_{010}$  selbst ja schon den Wert **0** hat.

(3) Untersuche der Reihe nach alle Belegungen, bei denen genau drei Variablen den Wert **1** haben, also in unserem Beispiel  $x=(1,1,1)$ . Alles Weitere läuft wie unter (0) bis (2).

### Feststellung 1:

Eine DNF kann  $2^n$  Vollkonjunktionen haben ( $n$  = Variablenzahl). Man braucht jedoch zur Charakterisierung oder/und Bearbeitung nie mehr als  $2^{n-1}$  Vollkonjunktionen zu betrachten, da man ohne Informationsverlust gegebenenfalls sofort zur negierten Funktion,  $\neg$ DNF, übergehen kann.

### Feststellung 2:

Es gibt Fälle, in denen die MLF trotz der eben getroffenen Feststellung 1 in bezug auf Speicheranforderung besser abschneidet als die DNF. Man betrachte z.B.  $(a \wedge b \wedge c) \vee d$ . Die negierte DNF benötigt 7 Vollkonjunktionen. Die MLF  $[(a \wedge b \wedge c) \vee d = abc + d - abcd]$  kommt mit 3 Bestandteilen aus. Die DNF braucht also (bei meinem oben gemachten Programmvorschlag) 7 Speicherzellen, die MLF 3 Zellpaare (je eine für den Bestandteilindex und eine für den Koeffizienten), also nur 6 Zellen.

### Feststellung 3:

Es können beliebig große Diskrepanzen auftreten: Im Ausdruck  $x_1 \vee (x_2 \wedge \dots \wedge x_n)$  benötigt die MLF immer noch nur 3 Speicherzellenpaare, also 6 Zellen. Die negierte DNF hat aber  $2^{n-1}-1$  Zellen nötig! Bei  $n=16$  sind das 32767 Zellen! Das ist kein Denkfehler. Die nicht negierte DNF benötigt 32769 Zellen! Und an dieser Diskrepanz scheint kein Weg vorbeizuführen (?) Die MLF ist in diesem Fall also wesentlich, ich betone wesentlich, besser als die DNF und die vorliegende Arbeit hat in eben solchen Fällen ihre volle Berechtigung.

### Feststellung 4:

Von einem wirklich intelligenten (einem "smarten") Forth-Definitionsword würde man dann noch verlangen, daß es die (beispielsweise für eine **IF**-Konstruktion) benötigte Bedingungsfunktion bei Bedarf entweder nach der DNF-Methode (siehe oben) oder nach der MLF-Methode compiliert, je nachdem, was bei der jeweils gerade vorliegenden booleschen Funktion günstiger ist.

### Offene Forderung zur Abrundung der Betrachtungen:

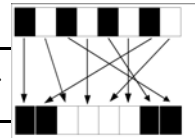
Zusammenfassend kann man sagen, daß die DNF verhältnismäßig einfach zu ermitteln ist (Funktionstafel), theoretisch jedenfalls, daß die MLF dagegen weniger leicht zugänglich ist. Was man plausiblerweise verlangen könnte, wäre ein Verfahren und ein Programm, das aus der DNF (in Tabellenform) heraus die MLF liefert. Ich möchte meine Anstrengungen hier beenden und es dem/der geeigneten Leser/in überlassen, ein solches Programm zu entwickeln. Wesentlich bei einem solchen Programm wäre die Entwicklung aller Kombinationen von  $k$  Einsen in einem  $n$ -Bit-Wort (stufenweise für  $k = 1$  bis  $n$ ). Ich habe ein Programm, das genau das leistet, in [3] und [4] veröffentlicht. Das besagte Programm ist in High-level-Forth gehalten und soll ganz nebenbei auch die Eleganz der Konstruktion **RECURSE** aufzeigen. Selbstverständlich würde man ein wirklich effizientes Programm zur Umsetzung der DNF in die zugehörige MLF vernünftigerweise als CODE-Definition fassen.

Auch wenn ich mich in der vorliegenden Arbeit bemüht habe, die MLF als eine zur DNF alternative Darstellungs- und Behandlungsmöglichkeit für boolesche Funktionen herauszuarbeiten, bedenke man doch, daß die DNF beispielsweise in der Zuverlässigkeitstheorie einen unschätzbaren Vorteil gegenüber der MLF aufweist: die Zuverlässigkeit eines  $n$ -komponentigen Systems läßt sich aus der DNF heraus sofort ablesen (siehe auch [1]). Man setze einfach die Komponentenzuverlässigkeiten in die DNF ein und ersetze Konjunktion durch Multiplikation, Disjunktion durch Addition und die Zuverlässigkeit einer negierten Komponente durch 1 minus der Zuverlässigkeit der betreffenden Komponente. Mit der MLF läßt sich das nicht machen.

### Literatur

- [1] Barlow, R.E. und Proschan, F.: Statistische Theorie der Zuverlässigkeit. Verlag Harri Deutsch, Fft/M. 1978.
- [2] Behringer, F.: Turbo-Forth, Conways "Life" und die DNF als IF-Bedingung. Vierte Dimension 15 (1999), H. 1.
- [3] Behringer, F.: Generating Combinations. Forthwrite, Heft 111 (2001).
- [4] Behringer, F.: Alle Kombinationen von  $k$  Einsen in einem  $n$ -Bit-Wort in High-level-Forth. Vierte Dimension 17 (2001), H. 2.
- [5] Böhme, G.: Einstieg in die Mathematische Logik. Carl-Hanser-Verlag München 1981.
- [6] Neumann, J. von: Probabilistic Logics and the Synthesis of Reliable Organisms from Unreliable Components. Annals of Math. Studies 34 (1956).





## Reorder und die Folgen

Martin Bitter

Ulrich Paul hatte in seinem Leserbrief in der Vierten Dimension 1/2000 ein von ihm entwickeltes Wort 'reorder' erwähnt, das den Umgang mit dem Stack sehr vereinfachen sollte. Ich bin also zu seiner Homepage gesurft und habe mir die Sourcen und beschreibende Texte geholt.

Was macht 'reorder'? Wie vereinfacht es den Umgang mit den Stackparametern?

Ein Beispiel: Ich weiß, dass auf dem Stack fünf Werte liegen, von denen ich die unteren drei brauche, diese noch in einer anderen Reihenfolge. Aus 1 2 3 4 5 -- soll 4 5 3 1 2 -- werden. Herkömmliche Lösung: ... 2swap swap 4 roll swap oder rot 4 roll 4 roll oder ... (im Wort) >R >R -rot ] Werte verbrauchen [ R> R> ... usw.

Die beste Lösung? Egal! Festzuhalten bleibt (zumindest) mir macht das Arbeit und da ich zur Sicherheit 'mal eben zwischendurch' teste, kostet es auch Zeit.

Nun kommt 'reorder'; das einfach aufgerufen wird und als Parameter den Inhalt der folgenden Klammer bekommt. Es parst die Klammer und generiert einen entsprechenden Code, der den Stack wie gewünscht manipuliert.

Also ... reorder ( 1 2 3 4 5 -- 4 5 3 1 2 ) ... das wars! In den meisten Fällen ist der generierte Code kompakter und schneller als die drei bis fünf ersetzten Stackworte.

Das sah gut aus! Das wollte ich auch! Aber: leider waren die Sourcen für F-PC verfasst und liefen unter Win32for nicht. Deshalb setzte ich mich hin und fing an die Sache zu analysieren. D.h. tief in die Quelltexte zu Win32for schauen, 486 Assembler-Kenntnisse aufzufrischen und zu versuchen, zu verstehen, wie 'reorder' bei F-PC genau funktioniert.

Ziemlich verwirrend war für mich die Tatsache, dass unter Win32for der TOS in einem eigenen Register gehalten wird. Nicht sosehr diese Tatsache war verwirrend, sondern ein wenig Benamungswirrwarr. Denn in (den spärlichen) Anmerkungen von Tom Zimmer dazu wird NOS einigemal als TOS bezeichnet, was ja stimmt, wenn man bedenkt, dass NOS der TOS des echten Stacks ist. (Ganz langsam und noch einmal lesen!). Als ich dadurch war, drang mir etwas ins Bewusstsein, das schon einige Zeit im Hinterkopf nagte: War ich nicht gerade dabei, das Rad noch einmal zu erfinden? Dabei etwas zu tun, das 'man' Forthern gelegentlich vorwirft?

Nun eine kurze E-Mail an Ulrich Paul klärte die Sachlage. Tatsächlich – jemand anderes hatte schon 'reorder' nach Win32for portiert oder besser gesagt, war gerade dabei. Das war die Antwort, die ich von Ulrich bekam: „Teufel auch! Da ist tatsächlich ein Bug drin! Aber die gute Nachricht: Es hat sich jemand mit dem Reorder befaßt und das auf Win32For portiert!“

Das war wirklich eine gute Nachricht: Heinrich Möller hat reorder portiert. Und dabei einen bisher unbekanntem Fehler im Original entdeckt, der daraufhin schnell behoben werden konnte.

Heinrich Möller ist sehr erfahren im Umgang mit Win32for und dem 486-Assembler. Ich konnte mich zurücklehnen und voller Genuss dem sich entwickelnden E-Mailverkehr folgen. Höchst interessante Detailfragen wurden diskutiert, unter anderem die Sache mit dem TOS-Register, ein Testprogramm geschrieben, das Stacks mit (theoretisch) beliebig vielen Einträgen erzeugt, 'reorder' mit allen möglichen Stackkombinationen aufruft und das Ergebnis überprüft. (NB. Das ist brute force lösbar – kostet aber mit steigender Stacktiefe immens Rechenzeit. Fred?!) Beide (Ulrich und Heinrich) feilten ein bisschen hier und dort - ich staunte. Das ganze war in ca. ein-einhalb Wochen 'gegessen'.

Das Ergebnis: ein reorder (natürlich statesmart) das unter F-PC und Win32for läuft, selbst fast keinen Platz im Wort belegt, sondern dort einen kompakten, schnellen Assemblercode erzeugt, der den Stack wie gewünscht manipuliert. Ich hab's ausprobiert. Eine tolle Sache! Als Nebenprodukt fiel ein definig word 'reorderer' ab: reorderer 3swap ( a b c d e f – d e f a b c ) erzeugt ein Wort '3swap' mit der angegebenen Funktion.

Mir hat nicht nur das 'Zugucken' Spaß gemacht. Oder das ich jetzt auch ein 'reorder' habe. Viel mehr als das, hat mich gefreut, dass hier Zusammenarbeit, Kollegialität und Austausch unter Mitgliedern, Gleichgesinnten deutlich wurde, wozu die Existenz der Vierte Dimension beigetragen hat.

Wer jetzt Lust bekommen hat 'reorder' selbst kennen zu lernen oder selbst am Code zu feilen oder das auch für andere Forthe nutzbar zu machen, Ulrich Paul und Heinrich Möller würden sich freuen: <http://www.paul.de/>

### Noch sind sie zu haben!

Einige wenige Exemplare der diesjährigen Tagungsgabe. Wer sich für die ganze Familie eindecken möchte oder wem gar ein Missgeschick passierte, sodass nun Ersatz benötigt wird, dem kann geholfen werden.



Eine kurze Anfrage an die Redaktion (sind noch welche zu haben?) und schon sind sie unterwegs. Die einzigartigen Swap-Tassen mit dem Logo der Forthgesellschaft und dem Schriftzug „Tagung 2001 – Dingden“ und einem 'süßen' Swappie..

Preis: 10 DM pro Tasse + Porto



*Dies ist der Vortrag, den Chris Jakeman auf der diesjährigen Tagung der Forth Gesellschaft e.V. vortrug. Er ist (natürlich) in englischer Sprache verfasst. Ich weiß, dass einige in der Leserschaft der Ansicht sind, dass in der Vierten Dimension veröffentlichte Artikel in deutscher Sprache zu lesen sein sollten.*

*Dies kann geschehen, da eine ausgezeichnete Übersetzung von Fred Behringer vorliegt. Aber das möge das Publikum entscheiden: Falls jemand, und zwar mindestens einer, dies wünscht, wird die deutschsprachige Version in der Vierten Dimension 4/2001 abgedruckt.*

MB

## WebForth

Chris Jakeman, FIG UK

**Abstract:** This paper presents an international project that tackles possibly the biggest obstacle to the continued use of Forth – attracting young people to work with Forth. Building on the premise that Forth is the easiest way to learn programming, the project delivers a web-site which encourages newcomers to programming to explore and develop programs in a convenient and immediate environment. The Forth language is made available at the user's browser through an applet.

**Origin:** The project had a long adolescence. In fact, technical difficulties (see below) delayed it almost beyond saving. Fortunately, and thanks to encouragement from Forth Gesellschaft, this paper announces the launch of the promised web-site, hosted at FIG UK (<http://forth.org.uk>).

Back in 1998, Mike Losh published "eForth for Java" in Forth Dimensions (Vol XX, No.1, May 98), a console-style applet which can be opened in any Java-enabled web browser. Also known as "jeForth", the source is copyright and released under

licence for free use for non-commercial purposes. "jeForth" is derived from eForth and Mike Losh extended it to support the examples from Leo Brodie's Starting Forth book. Through the development of "jeForth", Mike gained experience in using Java, but the primary aim from the start was "to create a simple Forth to demonstrate and promote over the Internet". Many Forths are freely available, but they all require the user to download and install software; "jeForth" eliminated this hurdle completely.

"jeForth" used to be published at <http://www.amsystech.com/~mlosh/> as show opposite. It is token-threaded and not the fastest possible implementation (cf. Misty Beach Forth at <http://www.mistybeach.com>). However it is quite small (36Kb), which is important when downloading from a web-site.

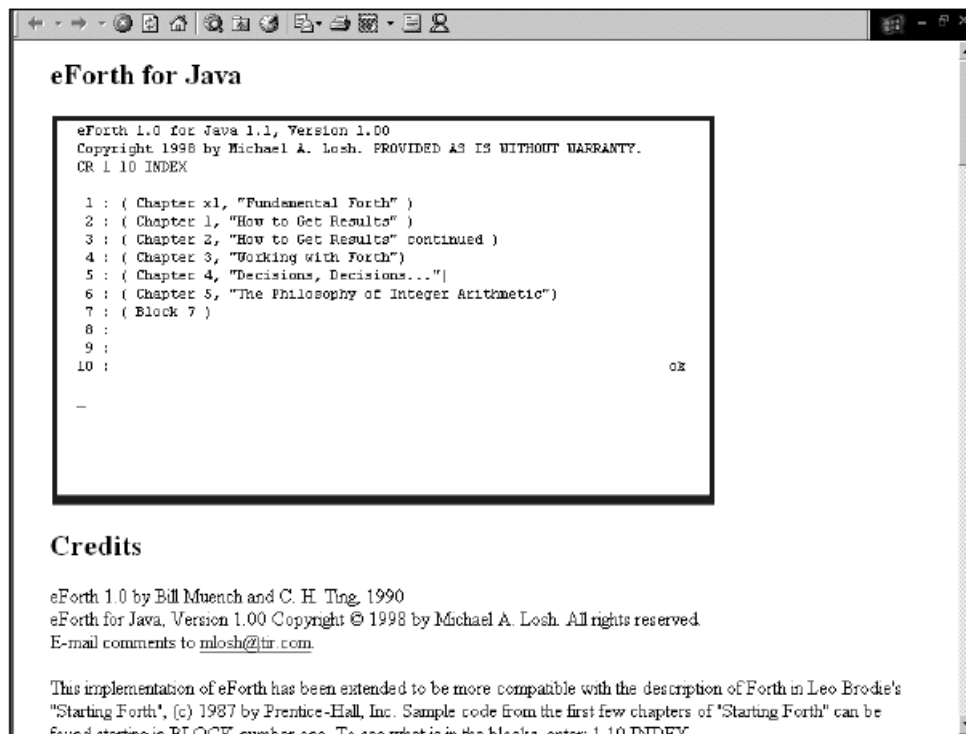
**Objective:** At FIG UK, we received an immediate response to Mike's appeal to "organize a project to develop a compelling web-site with a great tutorial for the jeForth applet".

As the software industry has matured, getting started in software has become harder than it used to be. C++ and Java have replaced the simpler C and BASIC languages, programming environments now come with multiple CDs of material to digest, and fewer people write software before they arrive at university. The teenagers who used to write programs are now assembling web-sites instead.

However the success of Lego Mindstorms has shown that young people can still enjoy creating programs. All they need is a simple but flexible environment.

Our aim for the WebForth project is to provide exactly that environment, supported by some short but effective tutorials.

**Teamwork:** After promoting the idea in the November 98 issue of the Forthwrite magazine, we were able to pull together a team with skills (or the willingness to acquire them) in Java, HTML, education and, of course, Forth itself. Fred Behringer was one of the first to volunteer and under his guidance we expanded our scope to provide a multi-language web-site, thus overcoming another critical barrier. The team listed here are all members of FIG UK. Since they span 6 countries in all, this project would have been impossible without e-mail.



Advisor	Mike Losh
Java	Philip Preston
HTML	Dave Pochin
Education	Graham Telfer
Translation	Fred Behringer
	Federico de Ceballos
Coordinator	Chris Jakeman



**Technology:** From the outset, we aimed to reach as many people as possible and not to rule people out because they were using an old browser, a slow modem connection or a small display screen. We took Internet Explorer 3 and Netscape Navigator 3 as the starting point and provided a zipped version of the site for people who prefer to explore offline.

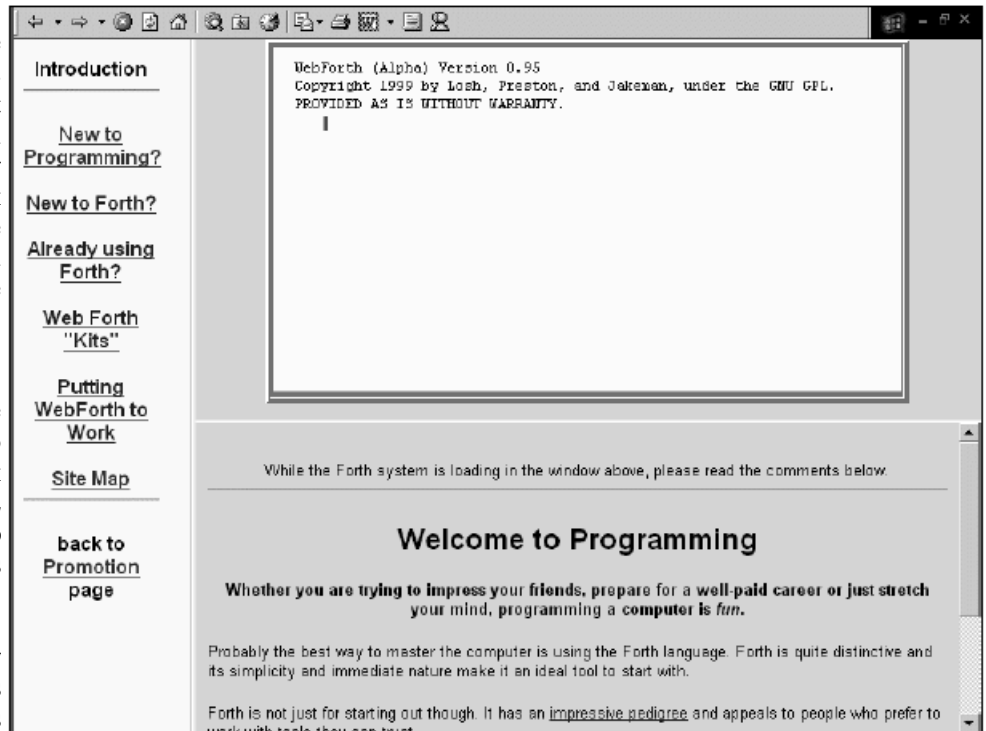
At the time, the Java libraries were still changing rapidly. We decided to stick to the JDK API v1.0.2 to work with older browsers. The HTML capability of browsers was also developing with Javascript, frames and cascading style sheets.

Of these candidates, we adopted only frames, an HTML feature that was over-used for a while, but allows us to keep the Forth console on the screen while scrolling an unlimited amount of tutorial text below it. The console is small enough to fit on an 800x600 display as shown opposite.

**Forth Developments:** One of the first improvements was to extend jeForth to comply with the ANS standard; successfully undertaken by Philip Preston and checked against John Hayes test suite. Initially the source code was kept in the traditional blocks and viewed with LIST and INDEX. Source code could be included in the HTML page and read by the WebForth applet. Philip extended it to read files as well. Applet security requires these files to reside on the host computer.

**Usability:** The first change for usability was to switch the dictionary look-up to case-insensitive operation. We also spent some time speeding up the text output – slow text leaves a very negative impression – and using colour to distinguish input, output and messages.

One unexpected issue turned out to be “focus”. It proved vital to indicate to the user whether it was the applet receiving keystrokes or the rest of the web page. For example, using the mouse to scroll down the page would remove the



focus from the applet, which would then appear to be dead.

Although running an applet within a web-page brings some usability problems (such as the focus), it also helps in some ways. When we switch from one page to another, the applet is re-loaded (usually from local cache) presenting a clean system for the user. In any tutorial system, it is necessary to protect the user from accidentally breaking his system (eg by typing >R at an empty prompt). We have collected a suitable set of checks to do this but have yet to implement them.

There is also a balance to be maintained between adding more functionality (eg access to the Java graphics library) and increasing the size of the applet which in turn extends the time to download.

**Multi-Language:** The entire WebForth site is available in English, German and Spanish and more languages can be added.

**Obstacles:** WebForth was ready for release in late 1999. Unfortunately, tests found too many systems which would not load the applet reliably. Once the applet was loaded, it ran well, but on some systems, re-loading the applet as the user moved from

```
<param name=cols value="72">
<param name=rows value="20">
<param name=fontsize value="12">
<param name=block1 value="">
<param name=block1.0 value=": SPACES 0 DO BL EMIT LOOP ; ">
<param name=block1.1 value=": AT-XY CR DROP SPACES ; ">
<param name=block1.2 value=": PAGE 20 0 DO CR LOOP ; ">
<param name=block1.3 value=": CONSTANT CREATE , DOES> @ ; ">
<param name=block1.4 value=": KEY? ?KEY ; ">
<param name=block1.5 value=": MS 0 DO 1000 0 DO LOOP LOOP ; ">
<param name=block1.6 value=": Welcome ( -- ) PAGE 29 0 AT-XY ." WELCOME TO FORTH" '>
<param name=block1.7 value=" CR CR CR CR CR CR 1000 MS ." To continue," '>
<param name=block1.8 value=" CR ." click this window and press any key" '>
<param name=block1.9 value=" ; ">
<param name=block1.10 value=": WelcomeLoop ( -- ) BEGIN Welcome 5000 MS KEY? UNTIL ; ">
<param name=block1.11 value=" ">
<param name=block1.12 value="WelcomeLoop ">
```



```

WebForth (Alpha) Version 0.94
Copyright 1999 by Losh, Preston, and Jakeman, under the GNU GPL.
PROVIDED AS IS WITHOUT WARRANTY.
OK
: TRY CR ." Welcome to WebForth " ; OK
TRY
Welcome to WebForth OK
|

```

page to page would cause Windows 95 and 3.1 to fail. Of course, any system published to promote Forth will do more harm than good if it appears to kill the user's computer.

However, as time has passed, these versions have faded in importance so full publication can now be considered and WebForth v1.0 is now available for use on-line or downloading for off-line work. I have Forth Gesellschaft to thank for providing the incentive to bring this work to a conclusion.

**Future Work:** WebForth already includes some tutorials – mostly simple programs with suggestions for extending them. We plan to add more. We intend to make WebForth fully crashproof and helpfully so too. We need a mechanism for on-line users to edit and store their trial programs at the host. And we need to attract users to the site.

Now that WebForth has finally been published, its success will depend on combining robust operation with tutorials at the

right level and a straightforward and clear environment. By encouraging and responding to feedback from our visitors, we aim to refine WebForth into a valuable service to the Forth community.



## Gehaltvolles Teil 1

zusammengestellt und übertragen von Fred Behringer

### VIJGEBLAADJE der HCC Forth-gebruikersgroep, Niederlande

**Nr. 25, April 2001**

**De belofte voor 2001 (2)  
Willem Ouwerkerk**

[<w.ouwerkerk@kader.hobby.nl>](mailto:w.ouwerkerk@kader.hobby.nl)

Die Elektronik des "Versprechens für 2001" (Roboter in Bau-

kastenform für den AT89C2051, für verschiedene AVRs und für das AF-Board). 6 Schaltbilder. IR-Signale auf einem modulierten 36-KHz-Träger. Ein Prototyp mit Namen Ushi läuft bereits - und fällt (IR-gesteuert) nicht vom Tisch. Ob es sich ein kleiner Club wie der HCC-Fgg leisten kann, einen kompletten Bausatz herauszubringen, ist noch nicht ganz entschieden - sagt Willem.

### Het ombouwen van een servo

**Willem Ouwerkerk**

[<w.ouwerkerk@kader.hobby.nl>](mailto:w.ouwerkerk@kader.hobby.nl)

Ein HS300 oder HS300BB (von Conrad) wird zu einem Roboter-Antriebsmotor umgebaut. Genaue Anleitung als "Kochrezept" in 15 Punkten.





## MINOΣ Example: OpenSched GUI

Bernd Paysan  
26. April 2001

### Zusammenfassung

Dieser Tagungsbeitrag zeigt, wie man mit MINOΣ einer etwas komplexeren Applikation ein Frontend verpaßt. Auch hier braucht sich Forth nicht in der Controller-Nische zu verstecken. Dieses Beispiel zeigt auch, daß ein Form-Editor nicht ausreicht (und nicht ausreichen kann), um eine typische GUI-Anwendung zu erschlagen.

### Einleitung und Motivation

OpenSched ist ein frei verfügbarer Projektplaner und -tracker. Wie viele Unix-Programme ist es zwar leistungsfähig, aber nicht „Boss-freundlich“. Die Eingabedaten stehen in einer einfachen Text-Datei. Der Autor hat hier, wie üblich, mehr auf einen einfachen Parser als auf Bedienfreundlichkeit geachtet. Es gibt zwar schon eine Projekt-Homepage für das zugehörige GUI, aber keinen Code.

Das allein reicht sicher nicht als Motivation, gibt es doch auch noch andere Projekte, bei denen die Situation ähnlich ist. OpenSched hat aber ein paar Eigenschaften, die es als Beispiel besonders nützlich machen:

Die Text-Datei ist einfach zu parsen, und zwar mit den Bordmitteln von Forth.

Die Eingabe hat eine leicht zu visualisierende Struktur. Auch gibt es nicht zu viele Möglichkeiten, das Beispiel bleibt übersichtlich.

Vor allem kommen tabellenartige Applikationen eben so häufig vor, daß es sich lohnt, hier einen typischen Vertreter herauszugreifen.

Das hier präsentierte OpenSched-Frontend ist noch nicht vollständig.

### 1 Das Hauptfenster

Bei der Projektplanung gibt es zwei wesentliche Größen: Tasks/Jobs und Ressourcen.

Am Bildschirm sieht das dann etwa so aus (siehe Abb. 2).

Um einen neuen Job anzulegen, tippt man einfach die ID in das Feld ein, und dann Return.

Genauso bei Ressourcen. Der Code dafür sieht so aus:

```
newjob get check-cr IF
new-job jobs append! list-jobs
s" " newjob assign THEN
```

Analog ist der Code für newresource. Einige Wörter hier sind in der Klasse opensched definiert, entweder als Variablen oder Methoden.

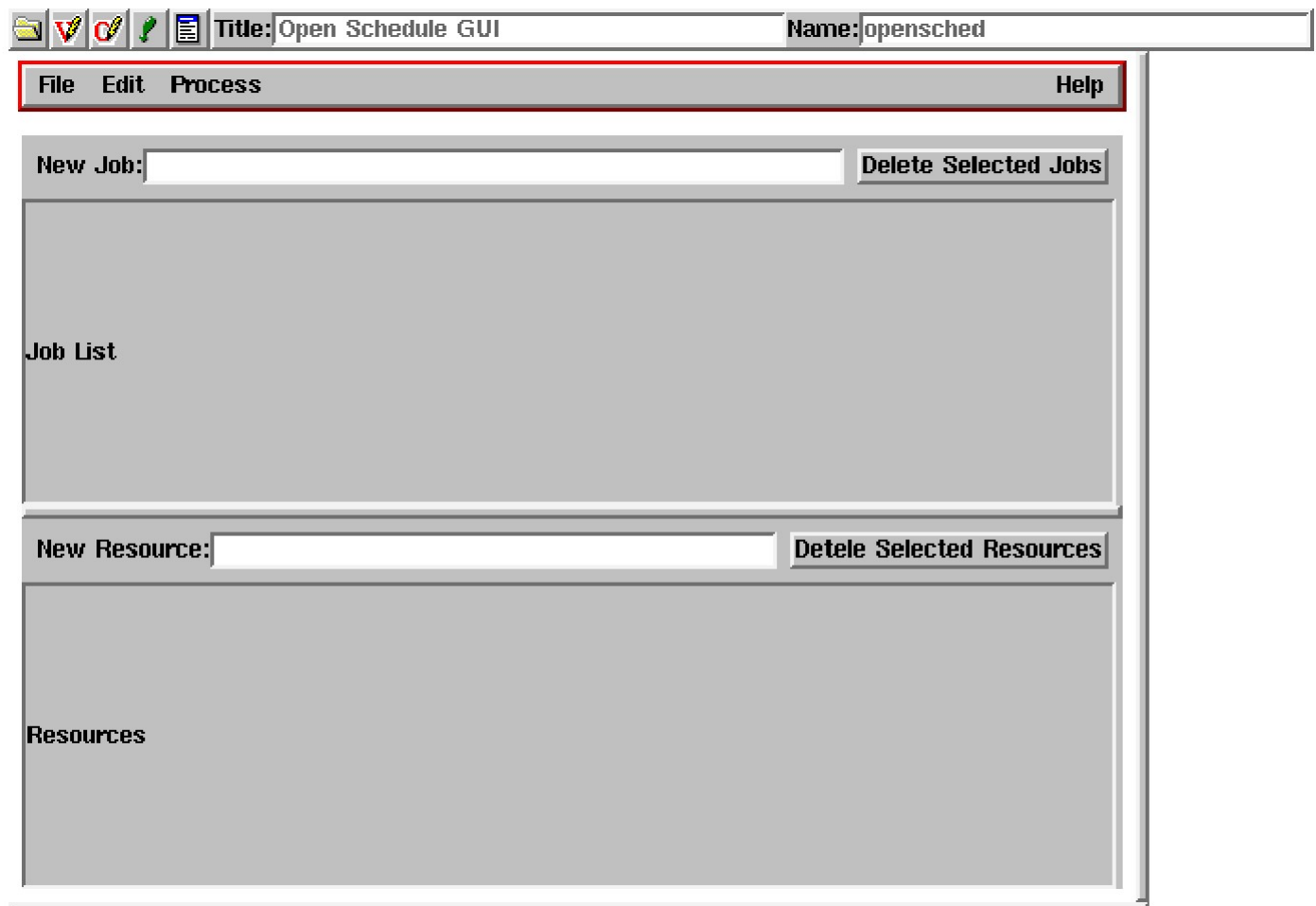


Abbildung 1 Main Window

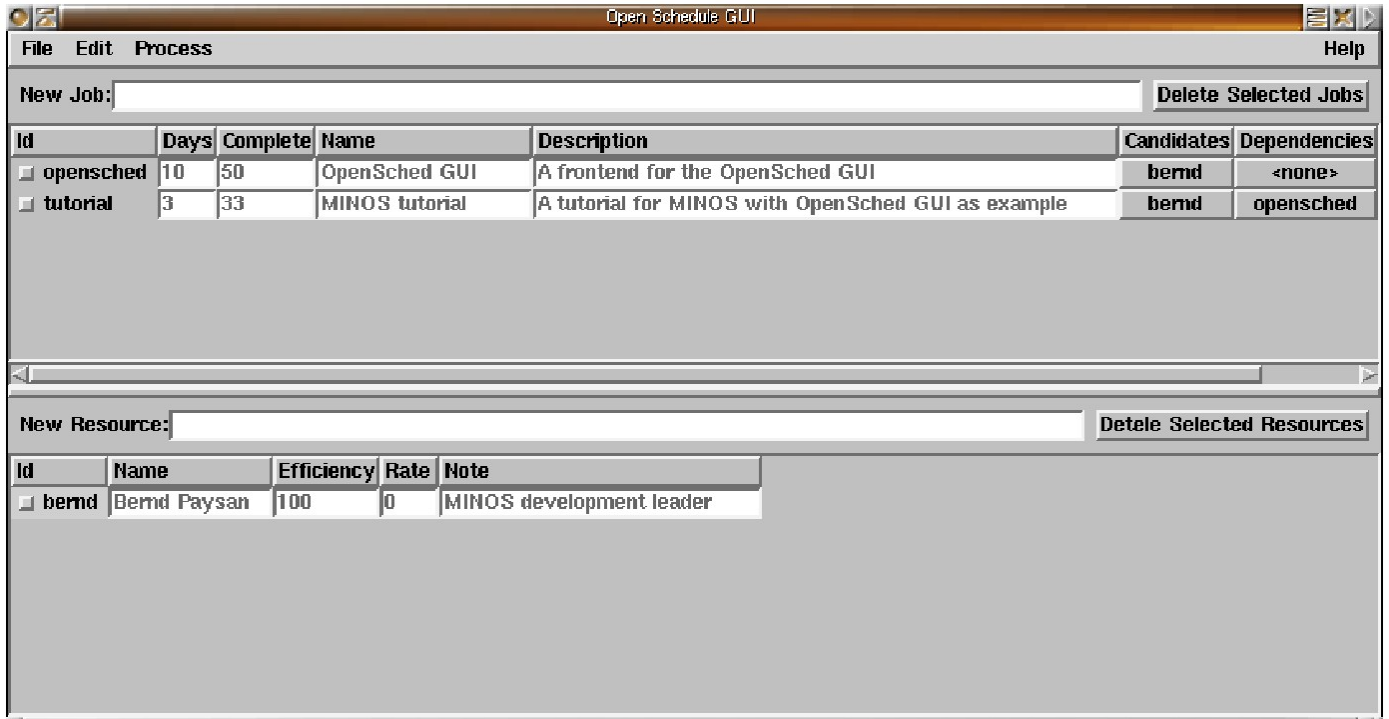


Abbildung 2 OpenSched Window

Die Variablen von opensched sind:

```

cell var resources
cell var jobs

cell var vacations
cell var milestones
cell var sched-path
cell var filename
sizeof global var globals
method list-resources
method list-jobs
method print
method load-sched

```

Die Implementierung der Methoden ist der eigentliche interessante Teil.

## 1.1 List Jobs

Zunächst müssen wir überprüfen, ob es denn überhaupt Jobs zum Listen gibt. Falls nicht, wird einfach ein leeres Feld angezeigt:

```

: list-jobs ( -- )
  jobs @ dup 0= IF
    drop s" Task List" text-label

```

```

new task-list assign EXIT
THEN

```

Für den Rest des codes ist also sichergestellt, daß es solche gibt. Wir erzeugen zunächst eine Titelleiste (Abb. 3).

```
>r job-title widget 1
```

Danach werden alle editierbaren Attribute eines Jobs in Tabellenzeilen gelistet. Die an die entsprechenden Buttons und Textfelder gebundenen Actions sorgen dafür, daß die jeweilige Variable modifiziert wird, wenn der Benutzer etwas eingibt.

```

BEGIN ^ TV[ r@ job selected ]T[ ]TV r@
  job id $@ tbutton new
^ IV[ r@ job days ]IV r@ job days
  @ 0 textfield new
^ IV[ r@ job complete ]IV r@ job
  complete @ 0 textfield new
^ VT[ r@ job name" ]VT
  r@ job name" $@
  textfield new
^ VT[ r@ job description ]VT
  r@ job description $@
  textfield new
0 r@ job candidate DT[ dpy dpy self
  with resources @ endwith
  ^ list! ]DT

```

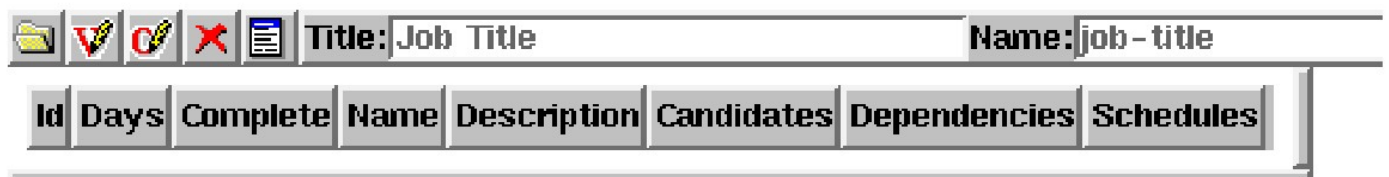


Abbildung 3 Job Title



```

r@ job candidate list-string
  button new
0 r@ job depends DT[ dpy dpy self
  with jobs @ endwith
  ^ list! ]DT
r@ job depends list-string button
  new
0 r@ job schedule DT[ drop ]DT
  s" edit" button new \ stub
$0 $1 *hfilll 2dup glue new
9 hatab new swap 1+
  ( --> box1..boxn n )
r> @ dup WHILE >r REPEAT drop

```

Zum Schluß noch etwas Glue dazu, und das Ganze der Job-Liste zuweisen.

```

$0 $1 *hfilll 2dup glue new swap 1+
vabox new task-list assign ;

```

Interessant ist hier noch, wie die Listen für Kandidaten oder Abhängigkeiten generiert werden. Der Benutzer selektiert dazu zunächst alle Jobs oder Ressourcen, von denen der Job abhängt, oder die als Kandidat in Frage kommen, und klickt dann auf die Kandidaten- oder Abhängigkeiten-Liste.

```

: scan-list ( head list -- head ) >r
BEGIN r@ job selected @
  IF r@ job id $@ new-list over
    append! THEN
  r> @ dup WHILE >r REPEAT drop ;

```

scan-list geht alle Elemente einer Liste durch, und erzeugt aus den IDs der selektierten eine neue Liste.

```

Variable list$
: list-string ( list -- addr u )
s" " list$ $!
BEGIN @ dup WHILE
  list$ $@len IF s" " list$ $+!
    THEN
  dup job id $@ list$ $+! REPEAT
drop
list$ $@ dup 0= IF 2drop s" <none>"
  THEN ;

```

list-string konvertiert eine Liste in einen String.

```

: list! ( head reslist button -- ) >r
over free-list scan-list list-string
r> button with assign endwith ;

```

list! erledigt den Rest, also die alte Liste löschen, die neue Liste anlegen, und den String in den Button bringen.

Das Generieren von Resource-Listen erfolgt analog. Die wichtigste Aufgabe ist damit eigentlich schon erledigt. Wir brauchen nur noch einen Weg, Dateien zu schreiben und zu lesen.

## 1.2 Dateien speichern

Eine Job-Beschreibung in OpenSched sieht etwa so aus:

```

task todo "Other things" 20
describe todo "Other things on the Todo list"
candidate todo bernd

```

Es kann dabei mehrere Kandidaten geben, außerdem kann der

Job schon teilweise oder ganz abgearbeitet sein, und natürlich kann der Job von anderen abhängig sein. All das packen wir in eine Struktur:

```

struct{
  ptr next
  ptr id
  ptr name"
  cell selected
  cell days
  cell complete
  ptr description
  ptr candidate
  ptr depends
  ptr schedule
  ptr nety
} job

```

Der Code zur Ausgabe eines Jobs braucht nicht ausführlich kommentiert werden.

```

: .id ( addr -- ) job id $@ type space ;
: .list ( list -- )
BEGIN @ dup WHILE dup .id REPEAT drop
;
: .job ( addr -- )
base push decimal
cr ." task " dup .id
" emit dup job name" $@ type
" emit space
dup job days @ 0 u.r
cr ." describe " dup .id
" emit dup job description $@ type
" emit
dup job complete @
?dup IF cr ." complete " over .id 0 .r
  THEN
dup job candidate dup @
IF cr ." candidate " over .id .list
  ELSE drop THEN
dup job depends dup @
IF cr ." depends " over .id .list
  ELSE drop THEN
cr drop ;

```

Neben den Jobs müssen noch globale Einstellungen und die Ressourcen ausgegeben werden. Wir verwenden hier die Umleitung von Ausgaben in eine Datei.

```

: print [ also fileop ]
filename $@ r/w output-file +buffer

```

Anschließend werden erst die globalen Einstellungen

```
globals .globals cr
```

dann die Ressourcen

```
resources BEGIN @ dup WHILE dup
.resource
  REPEAT
drop cr

```



und schließlich die Jobs ausgegeben.

```
jobs BEGIN @ dup WHILE dup .job REPEAT
drop cr
eot
```

Damit ist die eigentliche OpenSched-Datei erzeugt. OpenSched benutzt (unter anderem)  $\text{\LaTeX}$ , um die Reports zu setzen. Da OpenSched selbst keine Hauptdatei erzeugt, machen wir das.

```
filename $@ '/ -scan textmp $!
globals global prefix $@ textmp $+!
s" .tex" textmp $+!
textmp $@ r/w output-file +buffer
globals .tex eot
[ previous ] ;
```

Aus Platzgründen ersparen wir uns die anderen Print-Funktionen hier, so sehr unterscheiden sie sich nicht von der Behandlung eines Jobs.

## 1.3 Dateien laden

Ab Speichern allein reicht nicht, ein richtiges GUI-Frontend muß auch Dateien laden können. Das Format von OpenSched macht es uns einfach. Schließlich fängt jede Zeile mit einem Schlüsselwort an, die Daten folgen danach, Schlüsselwörter werden durch Leerzeichen getrennt. Die grundlegende Struktur des Parsers ist damit klar: wir definieren die OpenSched-Schlüsselwörter in einem eigenen Vokabular und laden die OpenSched-Datei einfach als (fast) normale Forth-Source.

Ein paar Hilfwörter brauchen wir noch. OpenSched verwendet ja keine strukturierte Eingabe (also kein XML), sondern vergibt „Attribute“. Nach jedem Schlüsselwort wird also die ID des Jobs oder der Resource angegeben. Die müssen wir also suchen:

```
: search-job ( addr u -- job )
current-widget @
opensched with jobs @ endwith >r
```

Die Variable `current-widget` enthält das aufrufende Hauptfenster. Darüber kommen wir an die richtige Liste. Nun müssen wir die einfach nur nach unserer ID durchsuchen.

```
BEGIN r@ WHILE 2dup r@ job id $@ compare
WHILE r> @ >r
REPEAT THEN
```

Sollten wir nichts gefunden haben, legen wir einfach einen neuen Job dieser ID an.

```
r@ 0= IF rdrop new-job dup
current-widget @
opensched with jobs endwith
append!
ELSE 2drop r> THEN ;
```

Damit haben wir alles beieinander, um ein Attribut zu definieren, etwa

```
: task job: >r r@ job name" add"
r> job days add# ;
```

Haben wir wirklich alles zusammen? Nein, Wörter, die Strings oder Zahlen parsen fehlen uns offensichtlich noch! Hier sind sie:

```
: add" ( addr -- )
" parse 2drop " parse rot $! ;
```

`add"` parst einen quotierten String und speichert ihn in eine Attribut-Variablen.

```
: add# ( addr -- ) base push decimal
bl word count s>number drop swap ! ;
```

`add#` parst eine Dezimalzahl, und speichert sie ab.

```
: add#2 ( addr -- ) base push decimal
bl word count s>number drop
dpl @ 0 max 2 min 2 ?DO &10 * LOOP
swap ! ;
```

`add#2` parst eine Fließkommazahl, und speichert sie als Prozentzahl ab.

```
: add-list ( addr -- )
BEGIN bl word count dup WHILE
new-list over append! REPEAT
. 2drop drop ;
```

`add-list` parst eine Liste von IDs, und fügt sie zu der Liste hinzu.

Die restlichen Attribute-Definitionen sehen ähnlich aus wie die oben, genaueres kann man in den Sourcen nachlesen. Nun müssen wir nur noch die Datei laden. Dabei sollte man berücksichtigen, daß OpenSched Kommentare mit ``#` abtrennt — leider diesmal ohne Space. Wir müssen das also direkt aus der Datei herausfiltern:

```
: sched-load ( addr u -- )
r/w open-file throw $200 input-file
BEGIN refill WHILE '# parse 2drop >in @
#tib ! >in off
interpret REPEAT
loadfile @ close-file throw ;
```

```
: include-sched ( addr u -- )
only previous scanner also
['] sched-load catch IF 2drop THEN ;
```

Das war im Wesentlichen das Grundgerüst für das Hauptfenster.

## 2 Globale Attribute

Ein klein wenig anders als die Jobs und Ressourcen behandeln wir die globalen Attribute eines Projekts. Das sind so Angaben wie der Dateiprefix der von OpenSched erzeugten Dateien, die Art der Dateien, oder (ganz wichtig!) das Startdatum. Dafür gibt's einen ganz normalen Dialog

Damit der Dialog den Status nach dem Öffnen auch anzeigt, wird entsprechender Code in der Show-Methode eingebunden:

```
: show ( -- )
current-widget @ opensched with globals
endwith >r
```

Zunächst weisen wir den Prefix-String zu:

```
r@ global prefix dup @ IF $@
prefix assign ELSE drop THEN
```



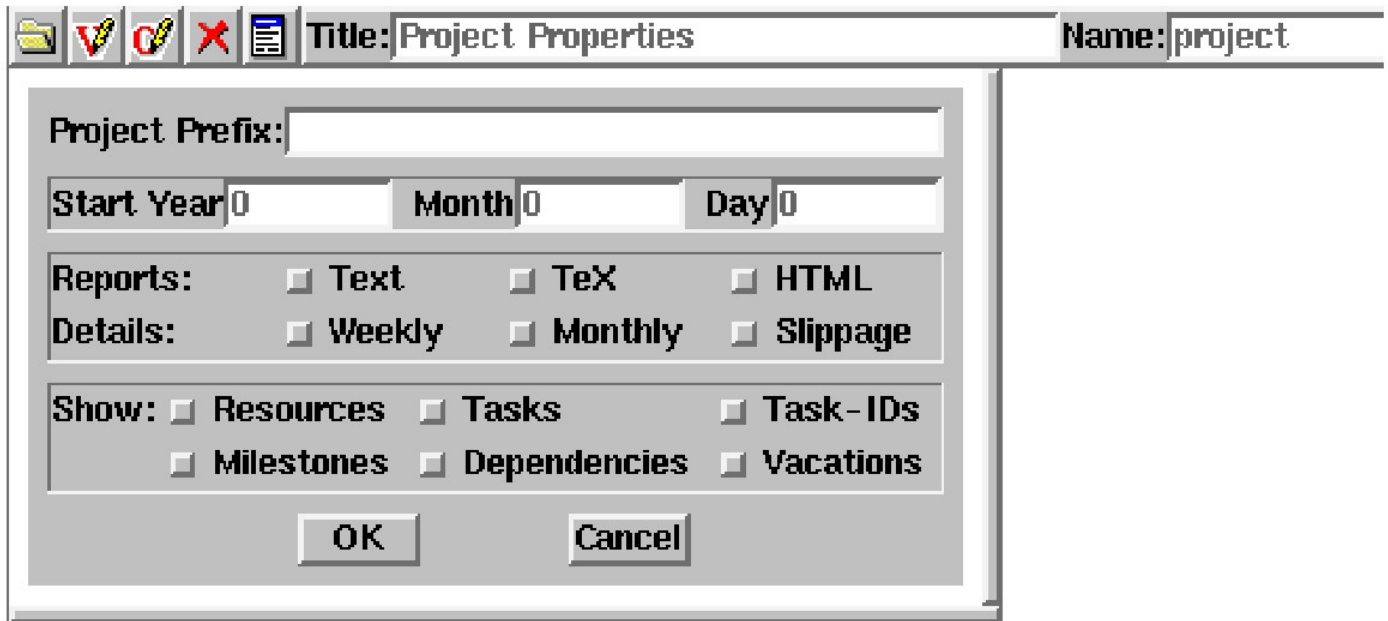


Abbildung 4 OpenSched Global Attributes

Das Start-Datum muß in Tag, Monat und Jahr zerlegt werden:

```
r@ global start @
&100 /mod &100 /mod
0 year assign 0 month assign 0 day
assign
```

Die Bits im Report müssen auch zerlegt werden:

```
r@ global reports @
dup 1 and IF text set THEN
dup 2 and IF tex set THEN
dup 4 and IF html set THEN
dup 8 and IF weekly set THEN
dup $10 and IF monthly set THEN
dup $20 and IF slippage set THEN drop
```

Ebenso wie die Bits in den Show-Flags:

```
r> global shows @
dup 1 and IF res set THEN
dup 2 and IF tasks set THEN
dup 4 and IF tids set THEN
dup 8 and IF mile set THEN
dup $10 and IF deps set THEN
dup $20 and IF vacs set THEN drop
```

Nun braucht man nur noch das alte Show aufrufen, dann wird das Fenster auch angezeigt.

```
:: show ;
```

Interessanten Code bekommt auch der OK-Knopf, der muß nämlich die ganzen Attribute abspeichern:

```
current-widget @
opensched with globals endwith >r
prefix get r@ global prefix $!
```

Den Prefix speichern wir einfach als String.

```
year get drop &100 * month get drop
+ &100 * day get drop +
```

```
r@ global start !
```

Das Datum speichern wir BCD-kodiert.

```
text get 1 and tex get 2 and or
html get 4 and or
weekly get 8 and or monthly get $10 and or
slippage get $20 and or
r@ global reports !
```

Die Flags für die Reports fassen wir zusammen (ein Bit pro Flag).

```
res get 1 and tasks get 2 and or
tids get 4 and or mile get 8 and or
deps get $10 and or vacs get $20 and or
r> global shows !
```

Genauso wird mit den anzuzeigenden Attributen verfahren.

```
close
```

Zum Schluß muß OK (wie Cancel) nur noch das Fenster schließen.

### 3 Die Menüs

Hier picke ich zunächst das interessanteste heraus, das Process-Menü. Wir können hier OpenSched, L<sup>A</sup>T<sub>E</sub>X und xdvi aufrufen, und uns die Ergebnisse ansehen. Der Aufruf von OpenSched sieht wie folgt aus:

```
^ opensched with
s" opensched " tmpbuf $!
filename $@ tmpbuf $+!
0 tmpbuf $@ + c!
tmpbuf $@ drop [ also dos ]
system [ previous ] drop
endwith
```

Die Aufrufe von L<sup>A</sup>T<sub>E</sub>X und xdvi sehen nicht viel anders aus.



Interessant ist auch der Code zum Datei laden:

```

^ opensched with
s" Sched File" s" " sched-path @
IF sched-path @$
  ELSE s" *.sched" THEN
^ S[ 2swap sched-path $! filename $!
^ current-widget ! load-sched ]S
fsel-action
endwith

```

Denselben Aufruf von fsel-action finden wir auch beim Abspeichern:

```

^ opensched with
filename @
IF print
  ELSE s" Sched File" s" " sched-path @
  IF sched-path @$ ELSE s" *.sched"
THEN
^ S[ 2swap sched-path $!

```

```

filename $! print ]S
fsel-action
THEN endwith

```

## 4 Ausblick

Das OpenSched-GUI unterstützt noch längst nicht alle Features von OpenSched. Es gibt auch noch GANTT-Charts und Resource-Charts, Milestones, Gruppen, Berücksichtigung von Urlaub, Kostenkalkulation, sowie die Möglichkeit, Arbeitszeiten an einem Job direkt einzugeben (vor allem für die Vergangenheit interessant. Man kann dann Plan und Wirklichkeit leicht ineinander überführen). Das alles wird mit der Zeit einfließen, und auch dieser Kurs wird entsprechend erweitert.

## 5 Appendix

Die von OpenSched im obigen Beispiel erzeugten Reports sehen wie folgt aus:

### 5.1 Time Frame

Start Date: 2001-04-20  
Finish Date: 2001-05-08

### 5.2 Task list

opensched OpenSched GUI  
2001-04-20 - 2001-05-03; assigned to Bernd Paysan

*A frontend for the OpenSched GUI*

tutorial MINOS tutorial  
2001-05-04 - 2001-05-08; assigned to Bernd Paysan

*A tutorial for MINOS with OpenSched GUI as example*

### 5.3 Milestone List

### 5.4 Resources And Task Assignment

#### 5.4.1 Bernd Paysan

Start	Finish	Days	Done	Task
2001-04-20	2001-05-03	10	5	opensched OpenSched GUI
2001-05-04	2001-05-08	3	1	tutorial MINOS tutorial

**Total: 13 days scheduled, 6 days completed.**

MINOS development leader

### 5.5 Dependencies

(2001-05-04 # 2001-05-08) MINOS tutorial

— (2001-04-20 # 2001-05-03) OpenSched GUI

### 5.6 Vacations

#### 5.6.1 Bernd Paysan

Start	Finish	Days

### 5.7 Month of April, 2001

Bernd Paysan

— 7 days - opensched - OpenSched GUI.  
(5 days completed.)

### 5.8 Month of May, 2001

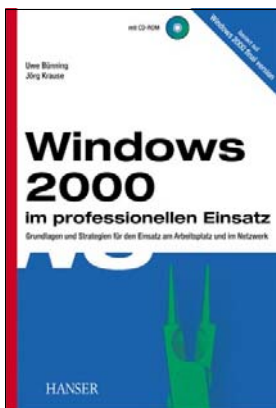
Bernd Paysan

— 3 days - opensched - OpenSched GUI.  
(5 days completed.)

— 3 days - tutorial - MINOS tutorial.  
(1 days completed.)

### 5.9 Gantt Chart





## Windows 2000 im professionellen Einsatz

Grundlagen und Strategien für den Einsatz am Arbeitsplatz und im Netzwerk

Uwe Bünning, Jörg Krause, Hanser Verlag, ISBN 3-446-21497-6

Einen ersten Update-Versuch von NT 4.0 nach Windows 2000 hatte

ich abgebrochen. Wie immer wenn es heißt, daß etwas ganz einfach funktioniert, ging bei meinem Rechner kurz nach den ersten Update-Bemühungen rein gar nichts mehr. Vor den erhofften Freuden am neuen System (Wozu eigentlich ? NT 4.0 arbeitet doch völlig problemlos ?) standen Probleme mit dem Dateisystem (FAT) das Windows 2000 nicht auf der Festplatte akzeptieren wollte, Partitionierungs-, Meinungsverschiedenheiten zwischen dem Setup und mir und Probleme mit Treibern zu diverser Hardware, die ich mir partout für mein System nicht ausreden lassen möchte.

Den nächsten Versuch, Windows 2000 bei mir zuhause ‚salonfähig‘ zu machen, wollte ich mit professioneller Hilfe durchführen, sprich mit dem o.a. Buch auf dem Schreibtisch. Immerhin spricht der Text auf der Buchrückseite davon, daß im Mittelpunkt des Buches die Einrichtung und Administration von Windows 2000 als leistungsfähige Workstation, im Peer-To-Peer Netz, als kleiner Intranet-Server sowie als Client in einem Windows 2000 Server Umfeld steht. Das ist doch genau das, was ich haben möchte.

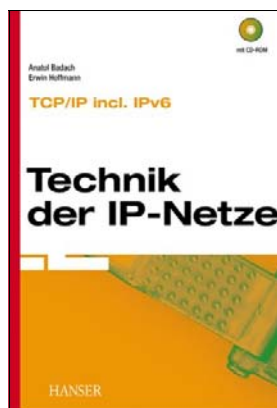
Da wundert es mich aber schon, daß die Autoren im Vorwort zu dem sichersten Betriebssystem der Welt 1,5 Seiten lang über das ‚Hacken‘ philosophieren, wobei ein Drittel dieses Textes als Auszug aus einem SF-Roman gekennzeichnet ist. Da heißt es am Ende des Vorwortes: „Wenn Sie Windows 2000 installieren, administrieren und anwenden, sollten Sie sich ein klein wenig wie ein Hacker fühlen.“

Und genau das möchte ich nicht ! Ich möchte das System installieren, für mein kleines, privates Netzwerk administrieren und dann einfach ‚laufen lassen‘. Dabei hilft mir natürlich immens die ‚Hackerinfo‘, daß ein Upgrade von Windows 3.x nach Windows 2000 generell nicht vorgesehen ist ;-)

Darüber hinaus ist das Kapitel über die Installation und die diversen Möglichkeiten Fehlentscheidungen zu treffen durchaus eine Hilfe, so wie das ganze Buch ‚im Grunde nicht schlecht‘ ist. An vielen Stellen sehe ich mich deutlich unterfordert, bzw. überfordert darin, Geduld zu üben und zum x-ten Male etwas zu lesen, was unzählige andere Autoren zuvor ebenfalls irgendwo abgeschrieben hatten.

Wer Implementierung und Administration von Windows 2000 in einem professionellen Umfeld durchführen muß, wird sich professioneller Hilfe bedienen [müssen] (z.B. des Microsoft Official Curriculum). Wer als Hobbyist Zeit und Lust (am Hacker-‚Feeling‘) hat, ist, sofern ihm/ihr die notwendige, weiterführende Literatur z.B. zu Netzwerktechniken z.V. steht, mit diesem Buch wirklich nicht schlecht bedient – aber auch nicht wirklich gut.

Friederich Prinz



## Technik der IP-Netze

TCP/IP incl. IPv6

Anatol Badach, Erwin Hoffmann, Hanser-Verlag, ISBN 3-446-21501-8

Auf hohem Niveau bietet dieses Buch eine fundierte Einführung in die Familie der TCP/IP Protokolle. Der Aufbau privater Netzwerke (VPN), Remote Access Services (RAS) und die Übermittlung multimedialer Daten (insbesondere Voice over IP) werden hier ausführlich behandelt. Zu jedem Kapitel stehen die praktischen Aspekte bei der Planung und Verwaltung von Netzwerken im Vordergrund, ohne die jeweils notwendigen Grundlagen zu vernachlässigen.

Das Buch ist sowohl als Lehrwerk für den Studierenden (Hochschule, Neu- oder Quereinsteiger in Netzwerktechnologie) gedacht, als auch als Nachschlagewerk für den Praktiker, der für die Planung, Realisierung, Verwaltung und Nutzung des Internets/Intranets z.B. eines Unternehmens verantwortlich ist.

Dieses Buch ist aus meiner Sicht rundum sowohl einer Empfehlung als auch den vom Verlag geforderten Preis von 89,- DM wert.

Friederich Prinz

Niederländisch ist gar nicht so schwer. Es ähnelt sehr den norddeutschen Sprachgepflogenheiten. Und außerdem ist Forth sowieso international. Neugierig? Werden Sie Förderer der

### HCC-Forth-gebruikersgroep.

Für 20 Gulden pro Jahr schicken wir Ihnen 5 oder 6 Hefte unserer Vereinszeitschrift ‚Het Vijgeblaadje‘ zu. Dort können Sie sich über die Aktivitäten unserer Mitglieder, über neue Hard- und Softwareprojekte, über Produkte zu günstigen Bezugspreisen, über Literatur aus unserer Forth-Bibliothek und vieles mehr aus erster Hand unterrichten. Auskünfte erteilt:

Willem Ouwerkerk  
Boulevard Heuvelink 126  
NL-6828 KW Arnhem  
E-Mail: [w.ouwerkerk@kader.hobby.nl](mailto:w.ouwerkerk@kader.hobby.nl)

Oder überweisen Sie einfach 20 Gulden auf das Konto 525 35 72 der HCC-Forth-gebruikersgroep bei der Postbank Amsterdam. Noch einfacher ist es wahrscheinlich, sich deshalb direkt an unseren Vorsitzenden, Willem Ouwerkerk zu wenden.



## Forth öffnet die Türen

Fred Behringer

<behringe@mathematik.tu-muenchen.de>

Einem Einsteiger erklärt man, was ein bedingter Sprung ist. Ein Umsteiger nimmt sich eine Forth-Befehlsliste und versucht, seine von anderen Sprachen her schon gefestigten Vorstellungen nach Forth hin umzusetzen. Hilfreich sind dabei kleine Programmstücke, an denen er sich ausrichten kann.

"Forth ist prädestiniert für Rapid-Prototyping", sagt Ulrich Paul in der VD 2/2001. Forth ist nicht nur Sprache, Forth ist auch System. "Forth ist eine Maschine", sagt John Jones in comp.lang.forth (wiedergegeben in der VD 2/2001). Was mich betrifft, ich verwende die virtuelle Forthmaschine auch gern zum schnellen und unkomplizierten Assemblieren. Auch das kann Forth, ohne Wenn und Aber.

Das untenstehende Programm hat 92 Bytes. Man könnte es ohne großen Aufwand mit DEBUG komponieren. Aber schon der einfachste 32-Bit-Registerladebefehl würde den Programmierer bei Einsatz von DEBUG unter DOS 6.2 in (unverständliche) Schwierigkeiten bringen: MOV AX,1234 geht. MOV EAX,12345678 geht nicht. Das ist in DEBUG nicht vorgesehen! Und die geschachtelten IF-ELSE-THEN-Konstruktionen, die in DEBUG ebensowenig möglich sind, die ich aber im untenstehenden Programm verwende, machen mir das hobbymäßige Programmieren im Forth-Assembler zur Freude.

Was habe ich vor? Kürzlich hatte ich die "Installations-CD" eines neu erworbenen CD-Laufwerks in der Hand. Das Schubfach öffnen und schließen: Ein Hilfsprogramm von 80000 Bytes! Nicht so wichtig, denn es gibt ja am CD-Fach den Knopf zum Schließen - aber wenn man bedenkt, daß das gleich folgende, mit Forth-Mitteln erstellte Programm nur 92 Bytes benötigt ... und der Knopf zum Schließen ist bei geöffneter Schublade unzugänglich.

Es wird aus Forth heraus eine unter DOS oder in einer DOS-Box von Windows-xyz ausführbare .COM-Datei erzeugt, die sich genauso schön aufrufen läßt wie die 80K-Datei von der "Installationsdiskette"! Getestet habe ich das auf Rechnern mit 486/66, K6/2/350 und Celeron/800 unter Turbo-Forth. Als Betriebssystem habe ich dabei DOS 6.2, DOS 7.0 (von der Windows-95-Notdiskette) und DOS 8.0 (sowohl von der Windows-98-Notdiskette als auch von der Windows-ME-Notdiskette) genommen.

Sicher gibt es für den Versierten verschiedene Mittel, von der Windows-95-Schreibtischplatte aus per Tastaturtastendruck das CD-Schubfach zu öffnen und dann auch wieder zu verschließen. Man braucht ja nur den CD-Player anzuwerfen. Mir hat es aber Spaß gemacht, das mit einer Sinnbildverknüpfung zu meinem untenstehenden kleinen Programm per Mausklick zu schaffen - "auch" zu schaffen, und das in Eigenarbeit ("selbstgemacht" - ich bin ständig am Üben), ohne das Ungetüm aus der Installations-CD bemühen zu müssen.

```

HEX
LABEL CDS                                \ "CD-SCHUB"
534D # 80 #) MOV                          \ "MSCD001" als ASCIIZ nach 80
4443 # 82 #) MOV                          \ (80-87 = Offsets in .COM-Datei).
3030 # 84 #) MOV                          \ Namensstring in CONFIG.SYS und
0031 # 86 #) MOV                          \ AUTOEXEC.BAT überprüfen!
      80 # DX MOV                          \ Geräte-Namensstringadresse nach DX
3D02 # AX MOV                              \ Geräteeinheit öffnen (lesen/schreiben)
      21 INT U>=                          \ Handle ist jetzt in AX
IF                                          \ Falls Fehler (CF=1), "Piep" und raus;
      AX BX MOV                          \ sonst AX nach BX
      5 # CX MOV                          \ Anzahl der zu lesenden Bytes
BYTE 6 # 80 #) MOV
      4402 # AX MOV                        \ CX Bytes aus Gerät BX nach DX-Adresse
                                          \ lesen
      21 INT U>=
IF                                          \ Falls Fehler, dann "Piep" und raus;
BYTE 1 # 81 #) TEST 0=                    \ sonst testen, ob geschlossen.
IF                                          \ Falls ja,
      0 # 80 #) MOV                        \ dann öffnen;
ELSE                                        \ falls nein,
      5 # 80 #) MOV                        \ dann schließen.
THEN
      1 # CX MOV                          \ Ein Byte an Gerät
      4403 # AX MOV                        \ mit Handle in BX senden
      21 INT U>=                          \ (Zwischenpuffer-Adresse noch in DX).
IF                                          \ Falls Fehler, dann "Piep" und raus;
      AX BX MOV                          \ falls fehlerfrei, dann
      3E # AH MOV                          \ Geräteeinheit mit Handle in BX
      21 INT RET                          \ schließen und raus.
THEN
THEN
THEN
      7 # AL MOV                          \ Piepton
      29 INT RET                          \ ausgeben.
CDS HERE SAVE CDS.COM                    \ Ab Labeladresse CDS als .COM-Datei
                                          \ speichern

```

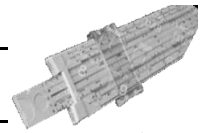
```

\ In Turbo-Forth als INCLUDE CDS einladen (Funktionstaste F3).
\ Die Zeilen bis zur vorletzten besorgen das Assemblieren
\ (ins Forth-Dictionary hinein). Die letzte Zeile (die ja beim
\ Einladen direkt ausgeführt wird) stellt daraus eine .COM-Datei
\ mit Namen CDS.COM (abänderbar) her. Es gilt:
\ Wenn die CD-Schublade geschlossen ist, öffnet "CDS [ret]" sie.
\ Wenn die CD-Schublade geöffnet ist, schließt "CDS [ret]" sie.
\ Tritt ein Fehler auf (Laufwerk oder MSCD001 gar nicht oder nur
\ unter anderem Namen vorhanden), so macht sich das durch einen
\ Piepton bemerkbar.

```

*Diese Rubrik ist neu in der Vierten Dimension. Fred Behringer folgt damit in löblicher Weise einer Anregung, die zum wiederholten Male auf der Jahrestagung geäußert wurde: Gezielt Umsteiger nach Forth und Anfänger in Forth anzusprechen und zu unterstützen. Ein Anfang ist gemacht! Die Redaktion wartet gespannt auf weitere Beiträge.*





*In der Vierten Dimension Ausgabe 2 2001 veröffentlichte ich einen Artikel aus „Spektrum der Wissenschaft“, voller Stolz, dass Herr Pöppe der Autor und verantwortliche Redakteur mir dies erlaubte. Umso größer war die Enttäuschung, dass eine falsche (roh-) Version abgedruckt wurde.*

*Spruchwort: „Dummheit und Stolz wächst auf demselben Holz!“*

*Unter anderem wurde diskutiert: Sonderauflage mit extra Post, Veröffentlichung einer Fehlerliste in der Vierten Dimension 3/2001 oder Neuabdruck.*

*Sie sehen, dass sich die Meinung Neuabdruck durchgesetzt hat. Der folgende Nachdruck enthält (hoffentlich, hoffentlich) alle vermissten Formatierungen, insbesondere Exponenten und mathematische Sonderzeichen (  $\Delta$   $\nabla$  ), die das Verständnis erleichtern.*

MB

## Rechnen mit garantierter Genauigkeit

**Die in die Computer eingebauten Rechenverfahren leiden unter Problemen, die im Extremfall die Ergebnisse völlig unbrauchbar machen können. Dies muss man beim heutigen Stand der Technik nicht mehr in Kauf nehmen. Fortgeschrittene Verfahren liefern Rechenergebnisse mit der Sicherheit eines mathematischen Beweises.**

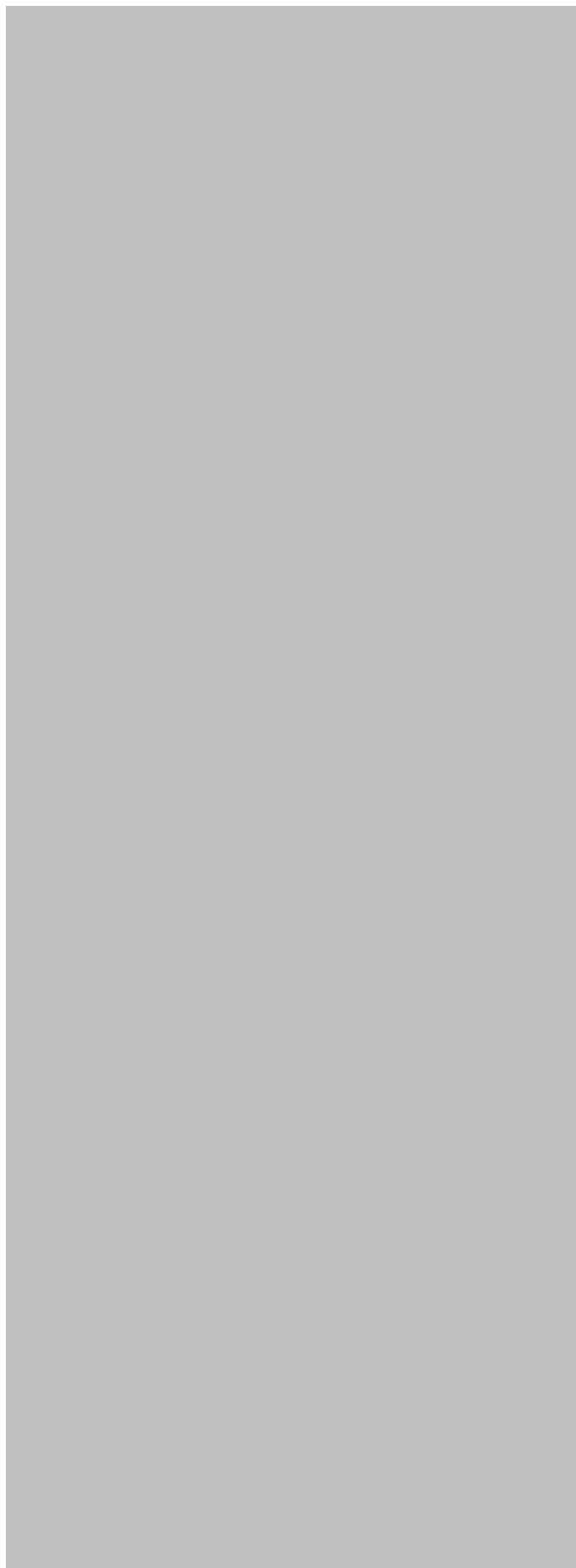
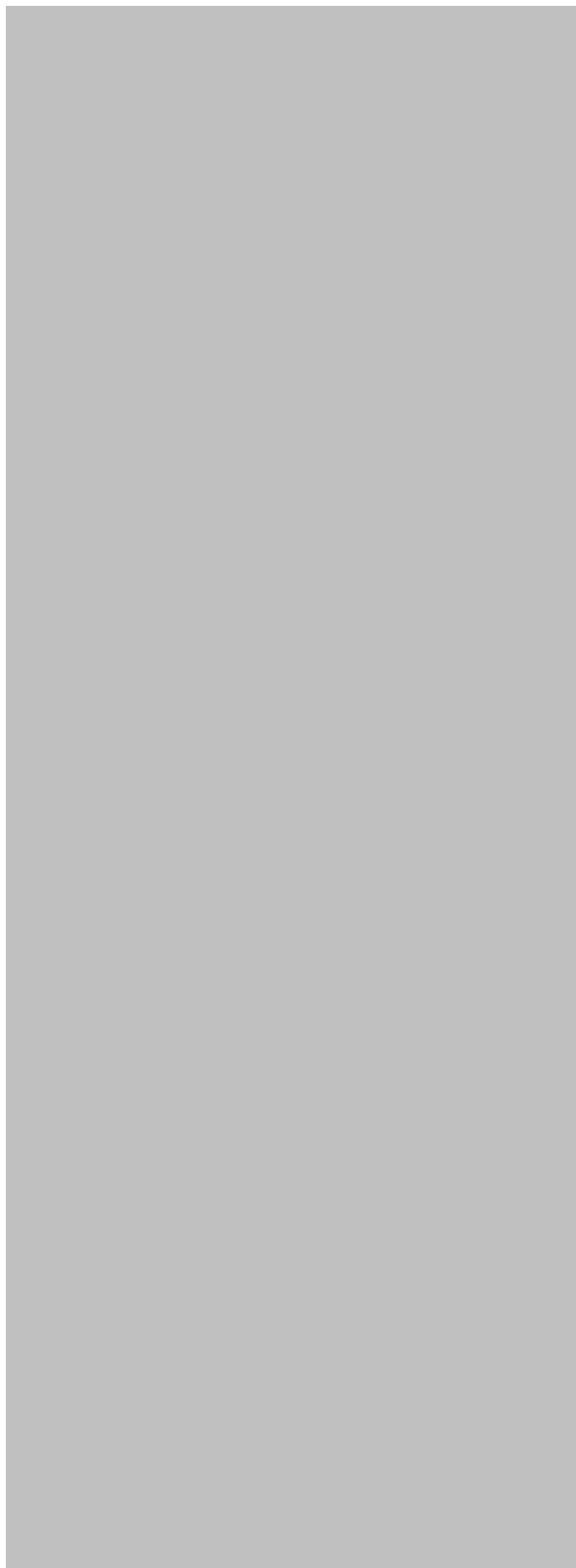
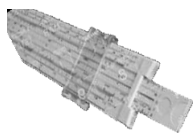
Von Christoph Pöppe

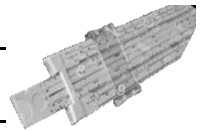
*Aus Spektrum der Wissenschaft September 2000,  
Seite 54*

Auf den Seiten 25 bis 30 fanden Sie in der gedruckten Version der Vierten Dimension den oben genannten Artikel von Christoph Pöppe. Wegen der Copy-Right-Absprachen mit dem Spektrum-Verlag können wir diesen Artikel hier nicht wiedergeben.

Wir verweisen stattdessen auf die On-Line-Version die unter <http://www.wissenschaft-online.de/abo/spektrum/archiv/5116> zum Download bereitsteht.

Ursprünglich war dies ein kostenloser Service, zur Zeit (Ende 2001) kostet dieser Download 1,25 €.





***FIGUK***

(Englische Forth-Gesellschaft)

Treten Sie unserer Forth-Gruppe bei.  
Verschaffen Sie sich Zugang zu unserer umfangreichen Bibliothek.

Sichern Sie sich alle zwei Monate  
ein Heft unserer Vereinszeitschrift.

(Auch ältere Hefte erhältlich)

Suchen Sie unsere Webseite auf:

*<http://www.fig-uk.org>*

Lassen Sie sich unser Neuzugangs-Gratis-Paket geben.

Der Mitgliedsbeitrag beträgt 12 engl. Pfund.

Hierfür bekommen Sie 6 Hefte unserer  
Vereinszeitschrift Forthwrite.

Beschleunigte Zustellung (Air Mail)  
ins Ausland kostet 20 Pfund.

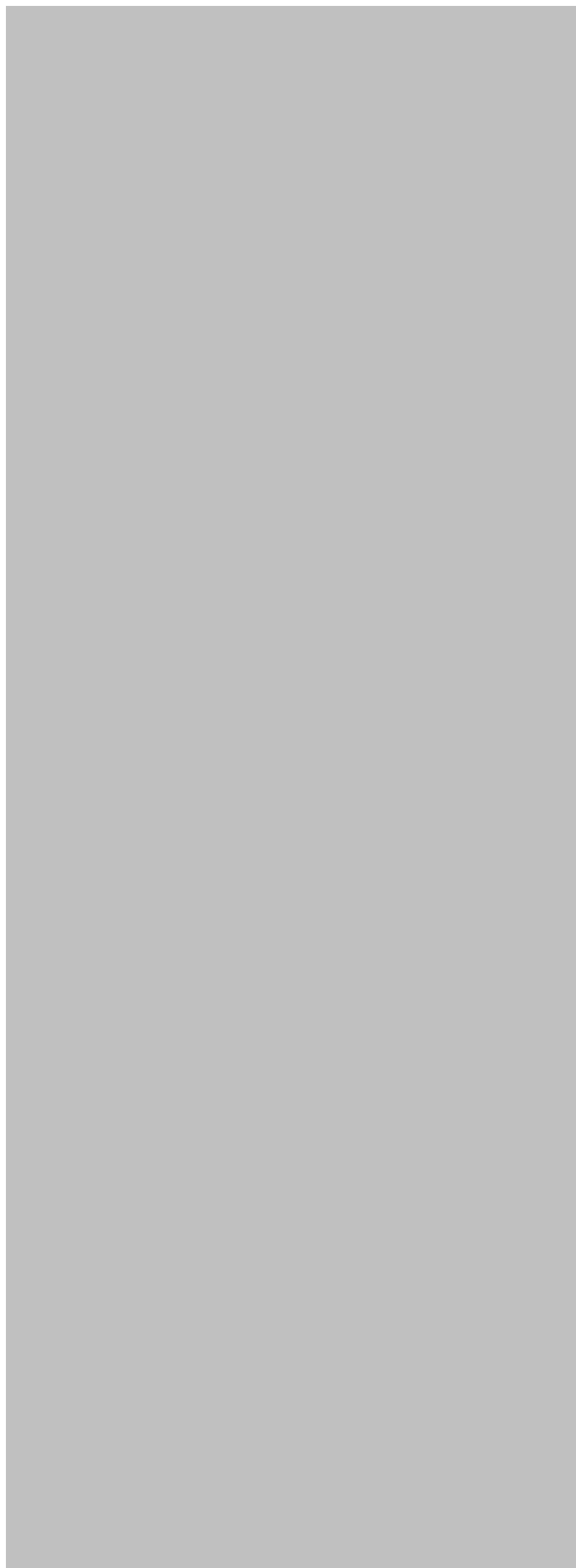
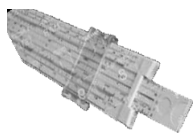
Körperschaften zahlen 36 Pfund,  
erhalten dafür aber viel Werbung.

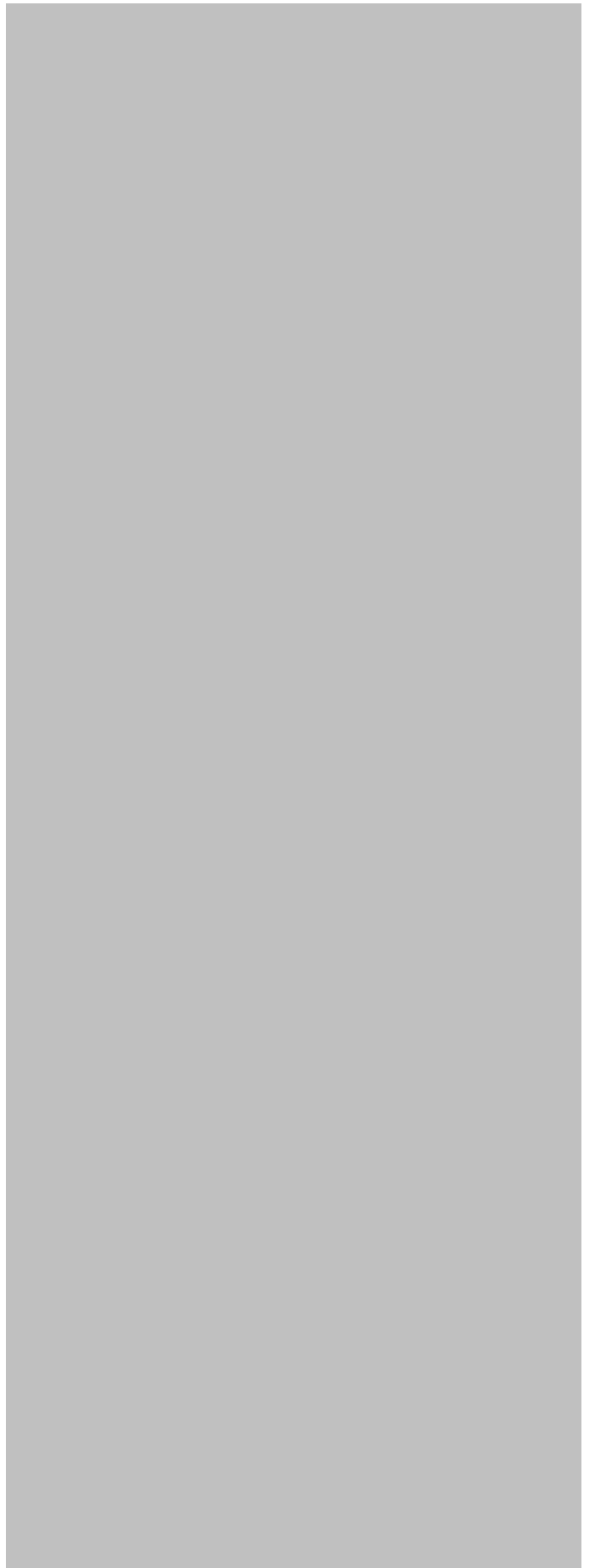
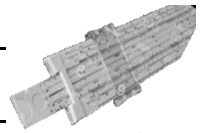
Wenden Sie sich an:

**Dr. Douglas Neale**  
**58 Woodland Way**  
**Morden Surrey**  
**SM4 4DS**

**Tel.: (44) 181-542-2747**

**E-Mail: [dneale@w58wmorden.demon.co.uk](mailto:dneale@w58wmorden.demon.co.uk)**









### Gehaltvolles Teil 2

zusammengestellt und übertragen von Fred Behringer

### 11 F11-UK

[jeremy.fowell@btinternet.com](mailto:jeremy.fowell@btinternet.com)

Genauere Angaben zum Hardware-Projekt: Etwa DM 200,- , HC11/8 MHz, Pygmy-Forth.

### FORTHWRITE der FIG UK, Großbritannien

#### Nr. 111 April 2001

Chris Jakeman begrüßt in seinem Editorial drei neue Mitglieder - aus London, Irland und Schweden. Er freut sich darüber, daß sich Leo Wong wieder schriftlich zu Wort gemeldet hat, und er begrüßt, daß Fred Behringer und Leo Wong mit ihren Artikeln starken Bezug auf den Permutationsbeitrag von Ed Hersom aus der Forthwrite 110 genommen haben. Er macht schon jetzt auf die Juni-Ausgabe aufmerksam, für welche er einen Bericht über unsere Forth-Tagung, an der er ja selbst teilgenommen hat und aus welcher er als neues FG-Mitglied hervorgegangen ist, schreiben wird.

#### 3 Forth News

**Dave Abrahams** <[d.j.abrahams@cwcom.net](mailto:d.j.abrahams@cwcom.net)>

John Hall und die neue FIG-US-Website; amrForth für den Motorola 6805/6808; Artikel von Julian Noble und von Chris Jakeman in SIGPLAN; Personal Forth Robot von Angelus Research; MacForth jetzt bei Mega Wolf Inc.; PFE 0.30; Anton Ertl und ein Peephole Optimizer für GForth.

#### 5 Six Easy Fonts: An Exercise in Win32Forth

**Dave Pochin** <[davep@sunterr.demon.co.uk](mailto:davep@sunterr.demon.co.uk)>

Windows hat (zumindest) sechs eingebaute System-Zeichensätze, die sich von Win32Forth aus über DC.f ganz leicht aufrufen und in eigenen Anwendungen verwenden lassen.

#### 8 F11-UK FIG Hardware Project

**jeremy.fowell@btinternet.com**

Checksums und Fehlerdiskussionen rund um das PygmyHC11-Projekt.

### 12 euroForth 2001

Vom 23. bis 26. November 2001 auf Schloß Dagstuhl bei Saarbrücken, organisiert von Dr. Bill Stoddart und Dr. Peter Knaggs.

### 13 Nominations for the FIG UK Awards - 2000

Aufruf zur Nominierung von Kandidaten für die beiden Preise für 2000: Förderung von Forth und Zeitschrift Forthwrite. Die Gewinner sind für ein ganzes Jahr vom Mitgliedsbeitrag befreit und ihre Namen werden in die Ruhmeshalle der FIG-UK-Website eingetragen.

### 14 Generating Combinations

**Fred Behringer** <[behringe@mathematik.tu-muenchen.de](mailto:behringe@mathematik.tu-muenchen.de)>

Wie bekommt man schnellstens und auf systematische Weise alle 16-Bit-Worte (im Computersinn), die alle ein und dieselbe vorgegebene Zahl von gesetzten Bits enthalten? Es wird mit RECURSE gearbeitet. Der Redakteur der Forthwrite: Ein paar der mathematischen Ausdrücke im zweiten Abschnitt verstehe ich nicht, aber die Bitmuster im Beispiel lassen erkennen, was geschieht und wie es geschieht.

### 17 Did You Know? und 18 Moving on

Chris Jakeman berichtet über die Eros-Raumsonde NEAR, deren Instrumente in Forth programmiert wurden (im nächsten Heft ein ausführlicher Bericht), erwähnt den Wegzug von Chris und Sylvia Hainsworth nach Spanien, begrüßt Jeremy Fowell als neuen Chairman der FIG UK und ist auf der Suche nach einem neuen Bibliothekar oder einer neuen Bibliothekarin, der oder die die Aufgabe von Sylvia übernehmen kann.



## 19 Vierte Dimension 1/01

Alan J M Wenham

<101745.3615@compuserve.com>

Ausführliche Besprechung unserer Zeitschrift. Alan verwendet neuerdings unser von Heinz Schnitter stammendes Logo. Bravo Alan! Und bravo noch einmal nachträglich, Heinz!

## 22 Dutch Forth Users Group

Diesmal wieder die Werbeanzeige der holländischen Forth-Freunde

## 23 Solving a Riddle

Leo Wong <hello @albany.net>

Ein "Rätsel", das aus den Newsgroups über Neuronale Netze etc von Steve Graham stammt: 5 Häuser, 5 verschiedene Farben, 5 Besitzer verschiedener Nationalität. Sie trinken bestimmte Getränke, rauchen bestimmte Marken und halten sich bestimmte Haustiere. Keine Überschneidungen, bestimmte Nebenbedingungen. Frage: Wer hält sich einen Fisch? Alle Permutationen durchzuprobieren, wäre eine unnötig komplizierte Aufgabe. Leo bietet ein Forth-Programm, das mit sehr viel weniger Aufwand auskommt.

## 30 Melbourne FIG

Die Forth Interest Group in Melbourne, Australien, stellt nur noch einen geselligen Club von zehn alten Freunden dar. Sie unterhalten sich jeden ersten Freitag im Monat über alles Computerbezogene. Über Forth haben sie seit zehn Jahren nicht mehr gesprochen. Vier von ihnen programmieren noch ernsthaft - in Delphi.

## 31 Letters

4 Briefe: Von Graham Telfer über editierbare Webpages (Wiki), von Andrew Holt über Java und die Zukunft von Forth, von Howard Oakford zu seinem Protokoll-Vortrag auf der euroForth und von Ian Thain über Forth, dessen Nichtbeachtung in der Welt der Programmiersprachen und warum er selbst trotzdem immer wieder auf Forth zurückgreift: Es macht ihm einfach Spaß.

## 35 Forthwrite Index

Eine Auflistung der seit 1990 erschienenen 375 Artikel der Forthwrite, nach Themen gruppiert. In Spalte 2 der jeweilige Autor, danach das Erscheinungsheft und schließlich der Titel. Auch elektronisch in verschiedenen Gruppierungsformen erhältlich (siehe <http://forth.org.uk>).

## Gehaltvolles Teil 3

zusammengestellt und übertragen von Fred Behringer

## VIJGEBLAADJE der HCC Forth-gebruikersgroep, Nederlande

## Nr. 26, Juni 2001

### Binair zoeken

Albert van der Horst

Binärsuche als Beispiel eines Algorithmus. Unabhängigkeit des eigentlichen Algorithmus von dem, was gesucht wird, als Beispiel eines Generischen Algorithmus. ALGOL, FORTRAN, Pascal, ADA, Java und in beschränktem Maße auch C++ sind abgesicherte Sprachen, Forth und C sind nicht oder wenig abgesicherte Sprachen (so der Autor). Generische Algorithmen lassen sich in abgesicherten Sprachen nur mit trickreicher Mühe programmieren.

### Cursus Forth, 9 juni

Auf den regelmäßigen Treffen wird auch immer ein etwa zweistündiger Kurs-Vortrag über Forth und die 8052-Hardware gehalten. Diesmal hier für die Nichtteilnehmer ein kleiner Ausschnitt. Das als Ersatz für den an dieser Stelle eigentlich beabsichtigten Bericht über den Fortschritt des Roboter-Projektes - sagt Willem Ouwerkerk.

### Das Forth-Treffen

fand am 9. Juni 2001 statt. Tagungsort ist weiterhin: Katholieke Vrouwengilde Nederland, Bionspoor 1204, Maarssenbroek. Die Website unserer niederländischen Forth-Freunde ist <http://www.forth.hccnet.nl>

## LEGO RCX-Verleih

Seit unserm Gewinn (VD 1/2001 S.30) verfügt unsere Schule über so ausreichend viele RCX Komponenten, dass ich meine privat eingebrachten Dinge nun Anderen, vorzugsweise Mitgliedern der Forth Gesellschaft e.V. zur Verfügung stellen kann! (Auszuleihen gegen Portokosten)

Angeboten wird: Ein komplettes LEGO-RCX-Set, so wie es für ca. 450 DM im Handel zu erwerben ist.

Inhalt:

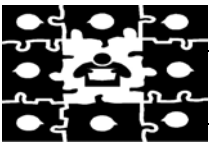
1 RCX, 1 Sendeturm, 2 Motoren, 4 Sensoren und ca. 1000 LEGO® Steine.

Anfragen über die Redaktion an mich.

Letztendlich enthält das Ganze auch nicht mehr als einen Mikrocontroller der Familie H8/300 Familie von Hitachi, ein 'paar' Treiber und 'etwas' Peripherie. Zudem: dieses Teil ist 'narrensicher'!



Martin Bitter



### Des Rätsels Lösung - Zahlendarstellung

Fred Behringer <behringe@mathematik.tu-muenchen.de>

Das habt Ihr richtig erraten: Die Zahl in der Vierten Dimension 1/2000 bezog sich auf die Schlacht bei Hastings, 1066, eines der wenigen Ereignisse aus der englischen Geschichte, das auch uns Deutschen geläufig ist. Der Halleysche Komet hängt mit ihr zusammen, und der Teppich von Bayeux. Der zweiten Zahl, der Binärzahl 10000101010, konnte man das leicht ansehen. Die erste Zahl, 10001111110, war von mir als 1066 in Negbinärdarstellung konzipiert (siehe unten), die dritte Zahl, 100000101000100, als 1066 im Fibonacci-System (siehe ebenfalls unten). - Das war's dann eigentlich.

**Erläuterung:** In der zweiten Zahl, einer Binärzahl, werden von rechts nach links, der Größe nach geordnet, die Koeffizienten einer Summe von Potenzen von 2 hingeschrieben. Das ist uns hinreichend bekannt.

In der ersten Zahl, einer Negbinärzahl, uns weniger bekannt, werden die Koeffizienten (auch nur 0 oder 1) einer Summe von Potenzen von -2 hingeschrieben, hier also:

$$1*(-2)^{10} + 1*(-2)^6 + 1*(-2)^5 + 1*(-2)^4 + 1*(-2)^3 + 1*(-2)^2 + 1*(-2)^1.$$

Die Anteile mit Koeffizient 0 habe ich hier weggelassen. Eine solche Darstellung habe ich vorzeiten mal zur Simulation eines A/D-Wandlers auf einem Analogrechner mit positiven und negativen Summierereingängen verwendet. Es war leicht, einen einmal hinzugefügten Widerstand an einem positiven Eingang durch einen anschließend eingefügten Widerstand halber Größe am negativen Eingang zu korrigieren. Es wäre aber etwas aufwendiger gewesen, einen einmal hinzugefügten Widerstand (ob am positiven oder am negativen Eingang) wieder wegzunehmen.

Die dritte Zahl ist 1066 in Fibonacci-Darstellung. Eine solche Darstellung bezieht sich auf positive Ganzzahlen. Auch hier habe ich Anteile mit dem Koeffizienten 0 weggelassen:

$$1*987 + 1*55 + 1*21 + 1*3$$

Eine Fibonacci-Zahl ist per definitionem die Summe ihrer beiden Vorgänger, angefangen mit 0 und 1 als nullte und erste Fibonacci-Zahl. In der zugehörigen Zahlendarstellung können nie zwei benachbarte Fibonacci-Anteile (mit Koeffizient 1) auftreten (als Koeffizienten kommen auch hier nur 0 oder 1 in Frage). Das beantwortet die im Rätsel gestellte Frage nach Einsparungsmöglichkeiten. Und außerdem bezieht sich die eben gemachte Aussage auch auf den Abstand von 0, weshalb die ersten beiden Fibonacci-Zahlen, 0 und 1, in der Fibonacci-Zahlendarstellung nicht verwendet werden, nicht verwendet zu werden brauchen.

Das Verhältnis zweier aufeinanderfolgenden Fibonacci-Zahlen nähert sich mit größer werdenden Fibonacci-Zahlen immer mehr dem Goldenen Schnitt ["(1 plus Wurzel aus 5) durch 2"]: Teile eine Strecke so, daß sich der kleine Teil zum großen so verhält wie der große Teil zum Ganzen. Dem Goldenen Schnitt begegnet man als übersättigter Massentourist in der prallen

Mittagssonne beim Lauschen der Ausführungen des Reiseleiters vor fast jeder der unzähligen europäischen Kirchen, deren Besichtigung auf dem Bildungsplan steht.

Addition, Subtraktion, Multiplikation und Division in Negbinär- und in Fibonacci-Darstellung in Forth wäre eine Herausforderung: Für den Geübten eine reine Übung, für den weniger Geübten ein willkommenes Studienobjekt, für den Anwender eine Bereicherung seiner Wortschatzkiste.

Ich hätte nie gedacht, daß es neben den mir vorschwebenden Lösungen noch andere geben könnte. Ich habe mir darüber aber auch nicht die geringsten Gedanken gemacht. Martin Bitter dagegen hat sich sehr viel Lösungsmühe gegeben. Es folgen Martins drei Zahlendarstellungen. Danke, Martin!

Hallo Fred,

nun ja ich weiß nicht, ob das 'deine' Lösungen sind, Lösungen sind es aber!

#### Lösung zu 10001111110 = 1066 !

Ein Rechnersystem arbeite nicht mit der Basis 2 sondern eben mit der Basis 3, Ich nenne das mal ternär. Seine Bits sind Tets (ternäre digits). Die Zahl T210 entspricht dann der Dezimalzahl 21 ( :-)>

Die Zahl Drei spielt eine besondere Rolle:

- jede Zelle ist drei Ziffern breit. D.h. in eine Speicherstelle passt maximal T222 = dezimal 26
- eine Zahl wird durch drei Zellen dargestellt und zwar in der Reihenfolge: High-Low-Mid (big-endian ist auch nicht besser aber trotzdem üblich).
- Es gilt die übliche Darstellung negativer Zahlen (höchstes Tet der Zelle gesetzt), d.h. alle Ziffern größer als 111 sind negativ, subtrahiert wird durch Addition des unechten 2er Komplements.
- Per Konvention liegen in der Mid- und High-Zelle ganze Zahlen, die Low-Zelle enthält die ersten drei Nachkommastellen (Fixkommadarstellung).

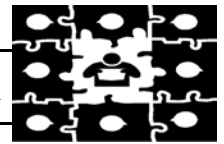
Das sieht dann so aus:

High	Low	Mid
000	000	001

Darstellung der Zahl 1

§ Zusätzlich gibt es eine 'floating-point' Darstellung mittels Exponent. Dazu wird eine vierte Zelle benötigt. In der High- und in der MID-Zelle liegen Ganzzahlwerte, in der Low-Zelle liegen die rationalen Werte.

Nun noch fehlende Nullen auffüllen und an einem Stück schreiben (so liegt es auch im Speicher!



Beispiele:

Exp	High	Low	Mid
000	000	000	001

Darstellung der Zahl 1

Exp	High	Low	Mid
000	000	001	000

Darstellung der Zahl T0,001  
(dez. 1/27 bzw. 0,370...)

Exp	High	Low	Mid
000	002	012	212

Darstellung der Zahl T2212,012  
(dez. 77 5/27 bzw. 77,1851...)

Nun 'deine' Zahl 10001111110.

Aufteilen in Zellen 010 001 111 110

Der Übersichtlicher erst mal ohne den Exponenten:

Exp	High	Low	Mid
	001	111	110

Darstellung der Zahl T1110,111  
(dez. 39 13/27 bzw. 39,481...)

Nun das ist ja noch nicht 1066, aber wir haben noch den Exponenten! Der beträgt T010 und das bedeutet: Nimm' das Ganze T010 mal mit T10 mal (verschiebe um T10 Stellen). T10 ist dezimal 3 also wird alles mit  $3 * 3 * 3 = 27$  malgenommen. Und sich da!  $39,481481481481... * 27 = 1066$

Exp	High	Low	Mid
010	001	111	110

Darstellung der Zahl T1110111  
(dez. 1066)

Ausführliche Überprüfungsrechnung:

$$T111 (\text{low}) = 13/27$$

$$T110 (\text{mid}) = 12$$

$$T001 (\text{high}) = 1 \text{ mal } T1000 = 1 \text{ mal } 27 = 27$$

$$T010 (\text{exp}) = 3, \text{ als Exponent der Basis (3) betrachten}$$

Daraus folgt:

$$1. \quad \text{addiere: } 12 \text{ plus } 27 \text{ plus } 13/27 = 39 \text{ } 13/27$$

$$2. \quad \text{Multipliziere das Ergebnis mit der Basis hoch Exponent: } = 3^3 = 27$$

$$\text{also: } (39 + 13/27) * 27 = 1053 + 13 = 1066 \text{ (qed)}$$

Das kleine Programm (Kasten) gibt zu jeder in dieses Schema passenden Ternär-Zahl zwei Zahlen aus, deren Quotient dieser Zahl in dezimaler Schreibweise entspricht.

Wann braucht man das? Ich weiß es nicht! Jemand der Leser? Aber: In der alten Vor-Rechner-Zeit, sortierten einige Menschen (ich!) Karteikarten mit einem 'Stricknadelsystem'. Das waren Löcher am Rand der Karteikarten, die man mit einer Schere in Schlitz verwandeln konnte. Um eine oder alle bestimmte(n) geschlitzte(n) Karte zu finden, reichte es nun aus, eine Stricknadel durch den ganzen Stapel Karteikarten zu stecken, den Stapel anzuheben und leicht zu schütteln: Was unten rauskam, waren die gesuchten Karten.

Meine alpha-numerischen Karten hatten zwei Lochreihen, das gab dann drei Möglichkeiten der 'Schlitzung': kein Schlitz, kurzer Schlitz, langer Schlitz. Und das ist ternär!

Ich konnte damals recht gut ternär zählen und buchstabieren :-)

### Lösung zu 100000101000100

(Hier ist dem Redakteur Gott-sei-Dank! ein Fehler aufgefallen, der weder dem Martin, noch dem Fred aufgefallen ist. Deshalb wird diese Lösung auch nicht veröffentlicht. So bleibt dem Martin und dem Fred eine Blamage erspart :-(Danke Redakteur!)

```

3 base !

: 3bfloat>s ( n n n -- n n )          \ wandelt Ternäre floating-pointzahl in einfachen Bruch um.
  1000 * +                             \ Wert der mittleren Zelle zu Low-wert addieren
  swap 1000 * 1000 * +                 \ High-Wert holen, mit Stellenwert malnehmen, addieren
  swap                                  \ Exponenten holen
  dup 0=                                \ ist er Null?
  IF drop 1000 EXIT THEN                \ Quotient = T1000
  dup 10 =                               \ ist er T10
  IF drop 1 EXIT THEN                   \
  dup 110 >                              \ und ist Exponent negativ?
  IF 100 - 1 swap                       \ Stellenwertfaktor bilden (110 + 10 )
  0 DO 10 * LOOP                         \ Quotient berechnen
  ELSE dup 10 >                          \ Exp. liefert garantiert keine Kommazahl
  IF 10 -                                 \ also
  0 DO 10 * LOOP                          \ Mantisse nach links schieben
  1                                       \ Quotient gleich Eins setzen
  ELSE 10 swap - 1 swap                  \ ansonsten
  0 DO 10 * LOOP                          \ entsprechenden Quotienten bilden
  THEN
THEN ;

010 001 111 110 3bfloat>s decimal / .

```



Der Bericht zu der diesjährigen Tagung der Forthgesellschaft wurde arbeitsteilig verfasst. Einen 'Stimmungsbericht' schrieb Friederich Prinz, einen eher 'protokollarischen' Bericht schrieb Bernd Paysan. Bernd übernahm diese Aufgabe dankenswerterweise recht kurzfristig. Er konnte daher nicht alle Beiträge erwähnen: Heinz Schnitter berichtet über Programmieren in verteilten System (... ein Forthrechner, der eine Nachricht an sich selbst nicht interpretieren kann, meldet sich nicht mit ok! sondern mit ? ..... ko!) und Ulrich Hoffman berichtete über die Feinheiten, die es bei der Programmierung eines 'digitalen Vier-Farb-Druckwerkes ( ca. 12 m lang, ca. 3 m breit ) zu beachten gibt.

## Forthtagung 2001 in Hamminkeln-Dingden, bei Wesel

Friderich Prinz

Martin Bitter hatte die diesjährige Forthtagung am rechten, unteren Niederrhein ausgerichtet. Das ist, für Eingeweihte, auf der falschen Seite des Rheins. Aber Dingden liegt in einer der reizvollsten Gegenden des Niederrheins, sehr ländlich, abseits großer Städte (die nahe Kreisstadt Wesel ist eher etwas verschlafen) und ca. 50 km nördlich des Ruhrgebietes. Dank der Tatsache, daß gerade NRW ganz hervorragend mit Autobahnen und Autobahnanschlüssen ausgestattet ist, hatte sich die Anfahrt der automobilen Tagungsgäste aber trotz des ländlichen Tagungsortes recht problemlos gestalten lassen. Wer am Freitag angereist ist, mußte allerdings ein wenig mit diversen Staus kämpfen, die vermutlich nicht zuletzt durch den großen Andrang zur Forthtagung, bzw. durch die in bemerkenswerter Vielzahl anreisenden Tagungsgäste selbst verursacht wurden. Zeitweise waren, einschließlich der begleitenden Damen, mehr als 30 forthige Gäste im Klausenhof versammelt.

Den bereits am Donnerstag angereisten Gästen hatte Jörg Plewe einen Rundflug in einem privaten Flugzeug in Aussicht gestellt. Der Rundflug, der am Freitag Vormittag stattfinden sollte, mußte – zum großen Bedauern der Gäste – wegen des heftigen Regens dieses Tages ausfallen. Statt dessen fuhren die Tagungsgäste nach Münster, wo es den Dom und die Altstadt zu besichtigen gab. Die Besichtigung des Doms war dabei der angenehmere Teil, weil trocken. Der Bummel durch die Münsteraner Altstadt fiel wegen des bereits angesprochenen Regens relativ kurz aus. Es schien allen Teilnehmern doch wesentlich angenehmer und gesünder zu sein, sich zu einem frühen Mittagessen in einem der vielen Restaurants niederzulassen. Immerhin bot aber der Blick aus den beschlagenen PKW-Scheiben auf der langen Fahrt nach Münster und zurück wunderschöne Ausblicke auf die regengraue, niederrheinische Landschaft.

Nach Murphy bessern sich auch die ungünstigsten Wetterbedingungen spätestens dann, wenn der geplante Aufenthalt im Freien planmäßig zuende geht. So war das selbstverständlich auch am Freitag. Bei der Rückkehr in Dingden, wo gleich weitere Tagungsgäste begrüßt werden konnten, rissen die Wolken auf, und es war endlich möglich, das weitläufige Areal des Klausenhofes in Gänze zu überblicken.

Die Tagung begann pünktlich. Martin Bitter stellte die bis dahin bei ihm eingereichten Vorschläge zu Vorträgen vor und

stimmte mit seinen Gästen die Reihenfolge der Vorträge ab. Unmittelbar danach begann das – wie immer – interessante Programm. Das hat allerdings einige Teilnehmer – wie immer – nicht davon abgehalten, sich spätestens nach dem gemeinsamen Abendessen leise aus dem Tagungsraum zu verabschieden und sich statt der Vorträge lieber dem Gespräch mit Freunden zu widmen. Der Heidekeller des Klausenhofes bot hierzu eine angenehme Atmosphäre und entsprechende Gelegenheit.

Der Samstag war geradezu vollgestopft mit Vorträgen und anschließenden Diskussionen zu dem jeweils gerade Gehörtem. Über Einzelheiten zu den Vorträgen wird Thomas Prinz gesondert berichten. (Die Skripte der Vorträge werden darüber hinaus in den nächsten VD veröffentlicht werden.) Vielleicht war es ein wenig zu viel des Guten. Die Vorträge wurden in dichter Reihenfolge gehalten und das kompakte Programm ließ wenig zeitlichen Spielraum für ‚Gespräche am Rande‘. Viele Teilnehmer kommen aber nicht zuletzt deswegen zu den Forthtagungen, weil diese ihnen die einzige Möglichkeit bieten, mit Vereinskameraden zu ‚parlieren‘, die man eben nur einmal im Jahr persönlich trifft.

Ohne einen der vielen, allesamt sehr guten Vorträge besonders hervorheben zu wollen, muß aber erwähnt werden, daß der Samstag mit einem Vortrag von Chris Jakeman begann. Die Direktoren der Forthgesellschaft hatten Chris, den Editor der ForthWrite, als besonderen Gast eingeladen. Chris ist sehr gerne gekommen und hat den Tagungsteilnehmern über das Web-Forth berichtet. Die Leser der VD haben davon bereits einiges aus den Berichten von Fred Behringer entnehmen können.

Der absolute Höhepunkt der diesjährigen Veranstaltung war der Programmier- und Bastelwettbewerb am Samstagabend. Martin Bitter hatte einen wesentlichen Teil eines Vehikels in vier einander gleichen Exemplaren vorbereitet; selbstverständlich mit LEGO-Bausteinen. Dieses Vehikel bestand aus einer bereits mit Rädern bestückten Achse, zugehörigen Motoren, einer Kugelschreibermine und natürlich dem LEGO-RCX. Das Vehikel sollte vervollständigt werden – wie auch immer – und auf einem Blatt Papier (DIN A 0) den Text „Forth 2001“ schreiben.

Die pfiffigsten Ideen mußte Martin allerdings gleich zu Anfang ablehnen. Morsezeichen, binäre ASCII-Darstellungen und Abkürzungen in römischen Ziffernzeichen waren nicht erlaubt. Trotzdem haben sich vier Gruppen mit großem Eifer auf die gestellte Aufgabe gestürzt und sich zur Erheiterung der Zuschauer sowohl in technische als auch in programmtechnische Irrungen verstrickt, von denen einige dazu geführt haben, daß allzu große, gegensätzliche Beschleunigen der angetriebenen Räder in selbsttätiger Zerlegung des Vehikels geendet sind.

Eine vollständige Lösung der Aufgabe konnte auch nach drei fröhlichen Stunden keines der vier Teams vorweisen. Darum mußte Martin die Fortschritte der einzelnen Teams danach beurteilen, in wie weit die bis dahin fertigen Teillösungen tatsächlich fähig waren, einzelne Buchstaben des geforderten Textes zu Papier zu bringen. Mit mehr oder weniger gutem Willen und einer gehörigen Portion Phantasie ließen sich dann auch einzelne Buchstaben erkennen, so daß zwei Siegerteams ermittelt werden konnten. Das Team „Rheinland“, bestehend



aus Michael Kalus und Söhnen, Jörg Plewe und dem technischen Berater Adolf Krüger, konnte ein Päckchen mit diversen, süßen Spezialitäten des Münsterlandes entgegen nehmen. Das Team Thomas Beierlein, Egmont Woitzel, Klaus Zobawa und Wolfgang Allinger hatte sich zu einem die nördliche Hälfte der Republik umspannenden „innovativen Viereck“ (Mittweida, Rostock, Hamburg, Solingen) zusammengefunden (siehe Photo). Dieses Team durfte ein Plüschtier mit nach Hause nehmen. Für den Süden Deutschlands hatte das Team „Bayern“, bestehend aus Heinz Schnitter, Jens Wilke und Bernd Paysan eigentlich einen Ehrenpreis verdient, weil es als Einziges auf die Idee kam, die ansonsten kaum kontrollierbaren Beschleunigungen des Vehikels mittels vorgeschalteter Getriebe zu bremsen. Und einen ganz besonderen Preis hätte Ulrich

Hoffmann verdient, der überwiegend als Einzelkämpfer an der gestellten Aufgabe gearbeitet hat. Ihm haben zeitweise Tochter Janna und Thomas Prinz assistiert.



Die insgesamt hervorragend organisierte Tagung (selbst die Tagungsbände lagen bereit !) hat einen zusätzlichen Höhepunkt durch die von Martin Bitter entwickelte und gefertigte Tagungsgabe erhalten, von der die Teilnehmer hoffen, daß sie zu einem Standard wird. Eine Tasse mit dem Swapdrachen (in Farbe !) und dem Aufdruck „Forthtagung 2001“ um das im letzten Jahr ausgewählte LOGO der FG herum ist natürlich etwas ganz Besonderes. Die Aufdrucke sind professionell gebrannt und entsprechend spülmaschinenfest. Davon möchte man mehr haben – zum Sammeln mindestens jeweils ein Exemplar der kommenden 24 Jahre !

Dingden war nach der Gründungsveranstaltung der FG in Kamp-Lintfort und nach der Tagung in Neukirchen-Vluyn die dritte und schönste Tagung am Niederrhein.

Vielen Dank, Martin !

Friederich Prinz

## Forthtagung 2001 in Hamminkeln-Dingden, bei Wesel

Bernd Paysan

Die Tagung fand dieses Jahr im Klausenhof in Hamminkeln-Dingden statt. Von München aus gesehen ist das noch hinter'm Ruhrpott und damit praktisch schon fast in Holland. Der Klausenhof selbst liegt am hinteren Ende von Dingden. Die einzelnen Gebäude wenden sich der Bezeichnung entsprechend mehr dem Innenhof zu, so daß Tagungssaal, Speisesaal, Bierkeller und Schlafstätten auf kurzem Weg zu erreichen sind.

Das landesübliche Wetter (eher schmutzelig) verhinderte am Freitag Morgen das Abheben des Tourismusprogramms in Jörg "Herkules" Plewes fliegender Kiste. Stattdessen besichtigten die früh Angereisten die Altstadt von Münster und den dortigen Dom.

Am späten Nachmittag begann dann das eigentliche Tagungsprogramm mit der **Forthbriefmarke** von Hans Eckes. Diese besteht aus einem PSC1000 (mit kleinem Kühlkörperchen), 128k SRAM, 512k Flash, SPI-Schnittstelle, RTC, UART und Watchdog, doppelseitig bestückt, links und rechts mit Pfostenstecker. Das paßt alles auf 2"x1". Die Briefmarke basiert auf dem Mini-Modul, das Hans letztes Jahr vorgestellt hat. Natürlich gibt es für die Briefmarke ein Forth (von Hans Eckes). Als Unterbau (damit man die Forth-Briefmarke auch benutzen kann) ist eine Platine mit Modellbau-Servos im Entstehen. Tester können wohl in Kürze so eine Briefmarke erwerben, und werden sich wohl nicht davon abhalten lassen, ihr eigenes Forth darauf zu portieren (so wie das beim Mini-Modul auch geschehen ist).

Die zweite Tagungsrunde startete am Samstag mit "special guest" **Chris Jakeman** von der FIG UK. Sein Thema: **Web-Forth**, ein Online-Tutorial für Forth, mit einem in Java geschriebenen Forth (ein eForth-Dialekt) als interaktive Übungsumgebung. Das Tutorial ist unter <http://forth.org.uk/> zu bestaunen. Für richtige Hardcore-Web-Surfer ist es natürlich wegen der "modernen" Techniken wie Java, JavaScript, Frames und Cascaded Style Sheets nicht nutzbar, aber die können ja schon Forth (oder? ;-).

Fred Behringer trug zum Thema "**Lego-Roboter und arithmetisierte Logik in Forth**" vor. Kurz gefaßt ist arithmetisierte Logik eine Methode, bool'sche Gleichungen so umzuformen und minimieren, daß am Schluß eine Summe gewichteter Faktoren herauskommt (wobei jeder Faktor 1 ist, wenn sein Term erfüllt ist, 0 sonst). Der Lego-Roboter, obwohl er "gestern noch funktioniert hat" tat sich dann aber schwer, dem vorgezeichneten Oval auf Papier zu folgen. Mit vereinten Kräften wurde der Fehler (zwei vertauschte Strippen) dann doch noch gefunden.

Der Vortrag von Klaus Zobawa ist leider nicht zum Abdruck freigegeben worden. Trotzdem: Vordergründig ging es um ein elektrisches Skalpell, das natürlich mit Forth gesteuert wird. Hinter den Kulissen zeigte Klaus aber, daß man sein "Thinking Forth" doch nochmal durchlesen sollte, und wie man die dort beschriebene Methode, ASCII-Rahmen-Tabellen in Forth so zu

verarbeiten, so verfeinern kann, daß damit ausführbare State-Maschinen entstehen. Da wirft sich schon die Frage auf, ob die übliche Meinung über Forth, es sei mehr für die Maschine, und weniger für den Programmierer, wirklich gerechtfertigt ist. Aber ein NEXT hat das Forth dann doch wieder.

Wolfgang Allinger beglückte uns gar mit zwei Vorträgen, "**MickerForth**" und seinen Erfahrungen zum **Quartus Forth**. MickerForth ist eigentlich gar kein richtiges Forth, jedenfalls hat es kein NEXT. Dafür ist der Stack richtig "spring-loaded", wie in Starting Forth; kommt ein neuer Eintrag dazu, werden die alten verschoben. Wolfgang setzt dieses MickerForth auf MickerCPUs (8051) ein, die zu klein für ein richtiges Forth sind, aber deren Berechnungen ihm zu kompliziert sind, um sie in Assembler zu codieren (lieber mit UPN). Mehr als die Taschenrechnerfunktionalität bringen die Macro-Assembler-Sourcen nicht mit, aber das erhöht den Komfort doch erheblich.

Wolfgang Allinger hat auch einen Weg gefunden, wie man die Yuppie-Nintendos (Palms) ernsthaft einsetzen kann, nämlich als handliches Terminal für die verschiedenen 8051 (mit MickerForth), die er in alle möglichen Geräte einbaut. Trotz aller Beschränkungen (MickerDisplay, kleiner Speicherplatz, Kratzstift und einer Schriftkennung, die bei den Forth-üblichen Sonderzeichen kneift) haben Palms nicht nur ein Forth, sondern auch andere Eigenschaften ausgewachsener Rechner (inclusive fehlerhaftes und lahmes OS ;-). Der Kampf mit den Tücken des Systems, des Forths, des Palm-Emulators und dem Auftraggeber hat sich aber gelohnt, das System erfüllt den Zweck.

Zum Abschluß des letzten Tagungsblocks gab's dann noch einen MINOS-Vortrag von mir (Bernd Paysan). Da es MINOS noch an ordentlicher Dokumentation fehlt, habe ich ein aktuelles Projekt herausgegriffen, und mit etwas Text versehen: Ein Frontend für **OpenSched**. OpenSched ist das freie Gegenstück zu M\$ Project, so wie LaTeX das freie Gegenstück zu M\$ Word ist (zugegeben, OpenSched ist noch recht jung). Als solches hat es natürlich ein Text-Interface. Das hält den Hacker nicht ab, aber welcher Hacker plant schon seine Projekte durch? Der Vortrag zeigte, daß MINOS und Forth zur Visualisierung solcher einfach zu parsenden Textwüsten sehr gut geeignet sind. Man kann nicht nur Bits beißen, auch Strings und GUIs gehen gut.

Das eigentliche Highlight der Tagung war aber der **Lechtzeit-Wettbewerb** im Anschluß an's Abendessen. Ziel war es, mit einem halbwegs vorbereiteten Lego-Roboter das Wort "FORTH" auf ein Blatt Papier zu bringen. Keiner der vier (sic!) teilnehmenden Gruppen glaubt, daß das mit der zur Verfügung stehenden Hardware vorher schon mal gelungen ist. Zunächst durften wir erst mal feststellen, warum Forth eine "schäbige, alte Sprache" ist (laut einem von Martins Lego-RCX-Büchern, C ist demzufolge "seit 30 Jahren kontinuierlich in Verbesserung"). Die Macken im System (pbForth) waren nicht gerade wenige:

- \* Die Übertragung ist nicht abgesichert. RCXe können sich gegenseitig stören, was einige Teilnehmer zur aktiver Sabotage, und alle Teams zum eifrigen "Hüttchen bauen" verleitete (Beste Hütte: Der Hut von Adolf Krüger). Und man muß genügend Leerzeilen einfügen, damit der RCX vor dem nächsten Befehl mit dem Compilieren fertig wird.

- \* Natürlich gibt's kein EMPTY, man muß vorher einen Marker setzen. Das merkt man natürlich erst, wenn der Speicher voll ist (neues Image laden).

- \* Auch einen "Notaus"-Knopf gibt's nicht - gegen die Endlosschleife hilft nur, die Batterien herauszunehmen (und dann wieder Image laden...).

- \* Das pbForth ist Novizen-unfreundlich case sensitive.

Dem allen möchten 20 Forther, die zusammen mindestens 40 Forthe geschrieben haben, natürlich am liebsten sofort Abhilfe schaffen.

Aber die eigentlichen Schwierigkeiten bereitet das Lego selbst:

- \* Die Motoren sind keine Schrittmotoren, sondern mit einer PWM angesteuerte Linearmotoren, und ein Tacho-Signal gibt's auch nicht. Schwung und wenig präzise Timer tun ein Übriges, um reproduzierbare Bewegungen zu vereiteln.

- \* Die großen Räder des Modells (statt etwa einer griffigen Raupe) und die geringe Untersetzung taten ein übriges dazu. Ein Getriebewechsel stellte sich als recht zeitraubend heraus.

- \* Natürlich bewegt sich die Turtle mit gesenktem Stift ganz anders, als mit gehobenem.



- \* Und da Lego nun mal nicht für bewegliche Teile gedacht war, flogen zwei der Modelle (eins beim Test, eins bei der Vorführung) mitten in der Fahrt auseinander.

Jedenfalls hatten wir viel Spaß damit, auch wenn es etwas Mühe kostete, das Gekrakel der jeweiligen Turtles zu entziffern. Hinterher hat man immer noch ein paar Ideen; am besten wäre es wohl gewesen, einfach 4th (ohne abzusetzen) zu zeichnen. Martin Bitter bekam noch die Aufgabe mit, eine befriedigende Lösung des Problems bei der nächsten Tagung vorzuführen.

Ob der Wettbewerb einen Einfluß auf die Entscheidung des anschließend geheim tagenden **Drachennrates** hatte, bleibt aber wohl für immer im Dunkeln, das dieses Zeremoniell umgibt.

*Bernd Paysan "If you want it done right, you have to do it yourself" <http://www.jwdt.com/~paysan/>*

## Protokoll der Jahresversammlung 2001 der Forth-Gesellschaft e.V.

*Sonntag, 29. April 2001 (Korrektur vom 7. Mai 2001)*

### Tagesordnungspunkt 1, 2 und 3: Begrüßung, Wahl des Schriftführers und Versammlungsleiters

Martin Bitter begrüßt die Versammlungsteilnehmer. Als Schriftführer wurde Jens Wilke berufen. Ulrich Hoffmann übernimmt die Aufgabe des Versammlungsleiters.

Versammlungsleiter: Ulrich Hoffmann  
Protokollführer: Jens Wilke  
Stimmberechtigte Mitglieder: 19

Der Versammlungsleiter stellt die Beschlussfähigkeit fest, da mehr als zehn Prozent der Vereinsmitglieder anwesend sind.

Zur Vollständigkeit seien die aktuellen Amtsträger aufgeführt:

Das Direktorium: Egmont Woitzel, Thomas Beierlein und Fred Behringer  
Das Forth-Büro wird von Ute Woitzel geleitet.  
Die Prüfung der Kasse: Wolfgang Allinger  
Das Internet wird von Egmont Woitzel betreut.

### Tagesordnungspunkt 4: Weitere Tagesordnungspunkte

Vorschläge für Verschiedenes werden aufgenommen.

### Tagesordnungspunkt 5: Bericht des Direktoriums

Thomas Beierlein berichtet über die erreichten Ziele. Die auf der letzten Tagung gesteckten Ziele konnten leider nicht vollständig erreicht werden. Thomas bittet die Vereinsmitglieder um ihre aktive Mithilfe, da die gesteckten Ziele ja nicht in erster Linie nur Aufgaben des Direktoriums sind.

### Ziele aus letzter Versammlung

Ausbau Internet, weitere VDs sind in das Internet gestellt worden.  
Der alte E-Mail-Zugang über die Kieler Box KBBS sollte abgeschafft werden, steht noch aus.  
Einsatz finanzieller Mittel: Finanzierung von einem Diplomanden wurde letztes Jahr gesponsert, leider hat sich niemand gefunden  
Forth-CD, technisch gute Fortschritte

### Kontakte zur holländischen Forth-Gruppe und anderen

Fred Behringer berichtet über ständigen Email Kontakt zu FIG UK und USA. Außerdem verweist er auf die ständigen Berichte in der VD über die amerikanischen Aktivitäten und das WebForth-Projekt von Chris Jakeman.

Vorschlag: Mehr Aktivitäten in Richtung der Holländer sollten unternommen werden. Zu Frankreich gibt es keinen Kontakt. Es gibt dort wohl keine "Forth Gesellschaft" mehr.

### Kontostand Stichtag 31.12.2000

Bankkonto: DM 34.160,73  
Postspargbuch: DM 373,85  
Insgesamt: DM 34.534,58

Weitere Vermögensgegenstände besitzt die Gesellschaft nicht.



## Einnahmeüberschussrechnung (Egmont Woitzel)

### Einnahmen

Mitgliedsbeiträge: DM 7.060,-  
Vierte Dimension: DM 6.037,41  
Jahrestagung 2000: DM 9.185,-  
Porto: DM 40,-  
Zinsen: DM 10,35  
Insgesamt: DM 22.638,49

### Ausgaben

Jahrestagung 2000: DM 7.605,50  
Internet (Mailbox (alleine DM 600) und Website): DM 898,12  
Porto (inkl. Versand VD): DM 2331,70  
Druckkosten VD: DM 1591,02  
Fracht VD: DM 269,00  
Bankgebühr: DM 109,46  
Projektausgabe LEGO: DM 500,-  
Stützung VD (ermäßigte Mitglieder): DM 96,-  
Sonstiges: DM 23,20  
Versand VD (Arbeitsaufwand Forth-Büro): DM 1173,00  
Verwaltung (Forth-Büro): DM 1357,48  
Insgesamt: DM 15.954,48

### Überschuss

Überschuss 2000: 6.684,01

## Zusammensetzung der Kosten

Egmont Woitzel macht die Einnahmen und Ausgaben des Vereins nochmal anhand von Kuchendiagrammen transparent, um einen besseren Eindruck zu vermitteln. Zum Beispiel: Der Versand der VD ist teurer als die Herstellung.

## Mitgliederentwicklung

Stand April 2001 hat der Verein 134 Mitglieder. Das sind 12 Mitglieder weniger als im Vorjahr. In den Vorvorjahren waren es jeweils 19, 34 und 41 weniger. Im Jahre 2001 gab es drei neue Mitglieder.

## Bericht Kassenprüfer

Wolfgang Allinger als Kassenprüfer bestätigt die ordnungsgemäße Verbuchung. Es ist aufgefallen, dass es einige Rückbuchungen aufgrund nicht mehr vorhandener Konten gab. Wolfgang empfiehlt, die angefallenen Kosten von den entsprechenden Personen einzutreiben. Außerdem empfiehlt sich zur Arbeitserleichterung der Einsatz eines besseren Buchhaltungsprogrammes.

Der Kassenprüfer empfiehlt die Entlastung der Buchhaltung. Die Tagungsteilnehmer entlasten das Forth-Büro für die Buchhaltung.

## Vorhaben des Direktoriums

Stabilisierung der VD: Thomas dankt Fritz und Martin für Ihren Einsatz und regt alle Mitglieder zur Mitarbeit an (schreibt Artikel).

Aktualisierung der Internetseite: Aufgrund seiner beruflichen Auslastung möchte Egmont die Pflege der Internetseiten abgeben.

## Tagesordnungspunkt 6: Entlastung des Direktoriums

Das Direktorium wird für seine Arbeit entlastet mit drei Enthaltungen, und keinen Gegenstimmen.

## Tagesordnungspunkt 7: Wahl des Direktoriums

Thomas Beierlein stellt sich für die Wiederwahl zur Verfügung. Er stellt klar, dass er das Amt nächstes Jahr (2002) aber nicht nochmal übernehmen möchte. Egmont Woitzel möchte aus persönlichen Gründen das Amt des Direktors nicht weiterführen.

Es herrscht darüber Einigkeit, die Wahl zu vereinfachen. Anstatt einzelner Personen werden die zwei Direktoren, die sich zur Wiederwahl stellen, geschlossen gewählt. Über das neue dritte Direktoriumsmitglied wurde pro Person abgestimmt.

Kandidaten: Bernd Paysan, Ulrich Hoffmann

Abstimmung: 6 Stimmen Bernd Paysan, 8 Stimmen Ulrich Hoffmann, 5 Stimmenthaltungen

Ulrich Hoffmann nimmt die Wahl an

Thomas Beierlein und Fred Behringer stellen sich zur Wiederwahl:

Abstimmung: 16 Ja-Stimmen für die Wiederwahl, 3 Enthaltungen

Thomas Beierlein und Fred Behringer nehmen die Wahl an

## Tagesordnungspunkt 8: Verschiedenes

### Internet

Die Betreuung der Internet-Präsenz übernimmt Ulrich Hoffmann. Fred Behringer stellt sich für redaktionelle Arbeiten zur Verfügung.

### VD

Friederich Prinz stand aus privaten Gründen leider nicht mehr zur Verfügung und bedankt sich bei Martin Bitter für die Übernahme dieser Aufgabe. Martin Bitter wird die nächsten zwei VD-Ausgaben betreuen. Friederich Prinz wird dann die Aufgabe wieder übernehmen. Wunsch: Es sollte eine Seite für die Anfänger eingerichtet werden, damit die VD nicht zum Insider-Blatt verkommt.

### Nächster Tagungsort

Wird von Heinz und Ulrike Schnitter im Raum München ausgerichtet.

Finanzielle Abwicklung und Versand der Tagungsbände wird vom Forth-Büro übernommen

Fred Behringer übernimmt die Organisation der Vorträge und Tagungsbände

Als Termin wird der 21.4. angestrebt (abhängig von der Hotelverfügbarkeit und Preise)

### Forth-Briefmarke

Vorschlag von Martin: Die Forth-Briefmarke von Hans Eckes sollte eventuell vom Verein "agressiv vermarktet" (Vorschlag Martin Bitter) bzw. unterstützt werden. Hans Eckes benötigt Beta-Tester für sein Projekt. Egmont Woitzel ist dafür, innerhalb der Forth-Gesellschaft Forth-Briefmarken zu erwerben und zum Testen zu verleihen.

Egmont Woitzel und Thomas Beierlein äußern Bedenken, bezüglich der "agressiven Vermarktung" aufgrund der Gemeinnützigkeit und steuerlichen Einstufung des Vereins. Egmont Woitzel bietet aber Unterstützung an, z.B. die Forth-Briefmarke unter "Projekte der Mitglieder" auf den Internetseiten des Vereins zu präsentieren.

### Buchführung

Laut Ute Woitzel und Egmont Woitzel besteht wegen des Buchhaltungsprogramms kein dringender Handlungsbedarf. Das bestehende Programm ist in der Lage, Euro zu verarbeiten und die Buchungen detaillierter (mit und ohne Mehrwertsteuer) aufzulisten.

Arndt Klingenberg schlägt vor, das Vereinsvermögen gewinnbringender anzulegen.

### Vermischtes

es sollte eine vollständige Mitgliederliste erstellt und verschickt werden

Egmont Woitzel sieht in der .NET-Strategie von Microsoft interessante Chancen und wird einen Bericht darüber in der VD veröffentlichen und eine Arbeitsgruppe ins Leben rufen.

München, 21.5.2001, Jens Wilke



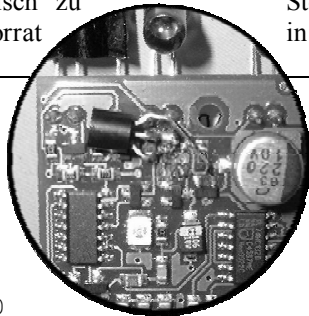


# Den Lego-Transmitter überlisten - mit und ohne Forth

Martin Bitter und Fred Behringer

"Den Finger drauf, das nehmen wir", sagte ein bekannter Feldherr. "Den Finger drauf, dann könnt' es gehen", sagte der eine von uns beiden. Beiden wurde nach zwei E-Mails klar: Damit war das Problem gelöst. Beide sind wir Hobbyisten. Experimentierfreudig. Uns verbindet, daß wir gern Probleme lösen - auch wenn wir die Mittel dazu nicht immer ganz beherrschen. Man kann nicht überall beschlagen sein. Im Beruf muß man sich ja auch ständig irgendwo "einarbeiten". Und in Forth macht das Einarbeiten ganz besonderen Spaß. Wenn etwas ohne Forth geht, ist es uns aber auch nicht unrecht. Und wenn etwas nach langer Anstrengung schließlich doch noch geht, quälen wir uns nicht unnötig mit der Frage danach, warum es denn nun geht, sondern schreiten zu neuen Taten. Claus Brod schrieb vor langer Zeit: „Öffne niemals eine Festplatte, es sei denn du bist so wohlhabend wie neugierig!“

Das Gleiche gilt für den IR-Tower. Es ist schon ein Risiko ihn aufzuschrauben und ihm elektrisch zu Leibe zu rücken. Natürlich hebt ein Vorrat



```

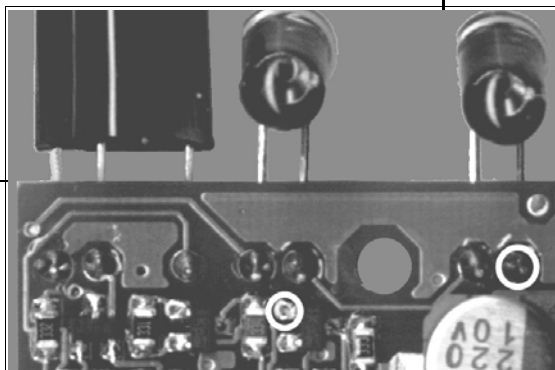
BASE @
HEX

: DUMP ( ad u -- )
  BASE @ >R HEX
  BEGIN
    DUP
    IF
      ABS 1- 10 / 1+ 8 MIN 0
      DO CR DUP DUP 0 <# # # # # # # # > TYPE SPACE SPACE
        8 0 DO DUP C@ 0 <# # # # # # # # > TYPE SPACE CHAR+ LOOP
    SPACE
      8 0 DO DUP C@ 0 <# # # # # # # # > TYPE SPACE CHAR+ LOOP
      SPACE SWAP 10 0
      DO DUP C@ 7F AND DUP 0 BL WITHIN
        OVER 7F = OR
        IF DROP [CHAR] . THEN
          EMIT CHAR+
      LOOP
      DROP
      LOOP
      BEGIN
        CR ." Weiter: Space   Zurueck: -   Raus:
      Return" CR
        KEY >R
        R@ 0D = IF 00 -1 ELSE 0 THEN
        R@ [CHAR] - = IF DROP 100 - 80 -1 THEN
        R> BL = IF DROP 80 -1 THEN
      UNTIL
    THEN
      DUP 0 =
    UNTIL
  DROP DROP CR
  R> BASE ! ;

BASE !

```

Ersatztowern diese Risikofreude erheblich. Man muss dann allerdings auch über Fähigkeiten verfügen, 'riesengroße' Ersatzteile zwischen all dem SMD-'Fliegendreck' zu positionieren (vgl. rundes Bild).



Der IR-Transmitter des Lego-Roboters hat die unangenehme Eigenschaft, nach jeweils 5 Sekunden, da er Signale zwar durchaus empfangen kann, selbst aber keine mehr sendet, die 9Volt-Betriebsspannung abzuschalten. Es werden im Betrieb nur 10 mA verbraucht, aber immerhin ist es bei längerem Betrieb sehr wohl angebracht, die Batterien zu schonen. Uns aber störte das. Wir möchten gern, daß der Transmitter in der Lage ist, eine beliebig lange Signalkette vom RCX aufzunehmen, ohne abzuschalten.

Wir fühlen uns aufgerufen, in bescheidenem Maße am pbForth mitzuentwickeln. Wenn wir hierzu DUMPen wollen, von Adresse soundso bis Adresse soundso, wird der RCX über den DUMP-Befehl angestoßen und sendet ab dann ein Byte nach dem anderen an den Transmitter und von dort an den Computer. Nach 5 Sekunden ist Sense. Rien ne va plus. Der Transmitter hat abgeschaltet. Das gefällt uns nicht. Schön wäre es, wenn man das Abschalten unterbinden könnte.

Man könnte ein "Herzschlag"-Programm verwenden (VD 1/2001, im Forthsystem entwickelt), mit dem der Transmitter interruptgetrieben alle 4,5 Sekunden einen "Anstoß" bekommt und für weitere 5 Sekunden in Bereitschaft bleibt. Das Programm aus Heft 1/2001 erzeugt aber mit jedem Anstoß ein Störzeichen auf dem Bildschirm. Das ist unschön. Wir dachten in mehrere Richtungen nach. Software und Hardware, Forth und nicht Forth.

Man könnte DUMP (WORDS wäre ein ähnlicher Kandidat) in Portionen aufteilen: 100h Bytes werden (wenn der zu dumpende Bereich überhaupt groß genug gewählt wurde) auf dem Bildschirm ausgegeben und dann wird gewartet, bis der Benutzer sich interaktiv für WEITER, ZURUECK oder RAUS entscheidet. Das im vorliegenden Artikel befindliche Programm, wohl das erste pbForth-Programm in der VD, erledigt das.

Wie war das nun aber mit dem Finger? Der Finger bewirkt, daß zwischen +9Volt und einer empfindlichen Stelle im Abschaltmechanismus des IR-Transmitters eine hochohmige Verbindung hergestellt wird, die das Potential nicht unter die Abschaltsschwelle sinken läßt und den Transmitter zum Weiterarbeiten zwingt. Wir rufen die Fachleute auf, uns in Reverse-Engineering einen Schaltplan zu rekonstruieren und - noch besser - uns zu erklären, wie es geht und warum. Vorerst freuen wir uns darüber, daß es überhaupt geht. Ein Bild der Towerplatine mit rot hervorgehobenem Layout (Handarbeit!) ist bei Martin Bitter (Redaktion) erhältlich

Zwischen +9Volt (rechtes Bein der rechten IR-LED) und dem aus dem Bild ersichtlichen Punkt an dem 33 K-Ohm Widerstand wurde über einen Schalter ein 2-Megohm-Widerstand gelegt. Der Schalter dient dazu, die Zwangsbereitschaft abzuschalten, wenn die Entwicklungsarbeiten beendet sind und die Schulkinder wieder auf die Lego-Roboter losgelassen werden.



Thomas Prinz schickt uns eine Übersicht, über das derzeitige Angebot an Mikrocontrollern (näheres sh. 3. Umschlagseite):

Es scheint so, als sei der Verleihservice etwas in Vergessenheit geraten. Wer jetzt merkt, dass er aus der Möglichkeit verschiedene Mikroprozessoren auszuleihen, seinen Nutzen, sei es beruflich oder einfach nur so aus Interesse, ziehen kann, der wende sich an Thomas Prinz.

**Verleihbare Forthmicros:**

**65F12 Rockwell 65F12 RSC-Forth**



Auf den Bildern kann (mit etwas Phantasie) rechts ein Diskettenlaufwerk erkennen.

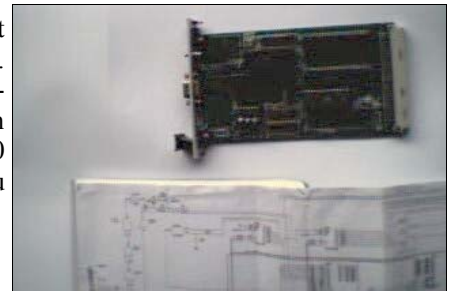
Es ist noch eine Beschreibung zu erstellen, da man das Laufwerk erst benutzen kann, wenn man einige Buffer per Hand initialisiert.

**Super8 Zilog Z8 Microcontroller mit Forth**



**8051\_xx bzw. 8052\_xx**

Immer noch mit BASIC im ROM. Geplant ist dies demnächst durch einen 8031 oder 80 C 320 von Dallas zu ersetzen.



Hierauf sollte eigentlich das Payne-Forth laufen, es braucht leider 16 k EPROM.

Die Adressdekodierung ist dafür nicht verdrahtet eine kleine Hardwareänderung ist notwendig.

**68F11 Der 68 HC 11 mit Forth von New Micros im ROM**



**Planung :**

Eine LED/LCD-Anzeige sollte sich von jedem Gerät ansteuern lassen.

Servos anzusteuern wie auf der I/O-Karte der FORTH-Briefmarke wäre natürlich auch eine Idee.

Ich beabsichtige noch einen AVR AT90S4414 umzurüsten. Man kann diesen einfach über ISP (In-System-Programming) Adapter von der Parallelen Schnittstelle aus programmieren.

Eine Forth wäre z.B. IRTC [www.ram-tech.co.uk/avr.htm](http://www.ram-tech.co.uk/avr.htm) der Spass kostet dort aber rund 200 Britische Pfund 8-((

Evtl. reicht aber auch die Software von ATMEL um das Teil ganz konserativ über Assembler zu programmieren. Etwa wie das Micker-Forth von Allinger. \*

Tschüss und noch einen schönen Tag

Thomas Prinz

\* Beitrag erscheint in der vierten Dimension 4/2001



## From the big Teich

### From the big Teich ...

Henry Vinerts

*Dieser Bericht unseres langjährigen 'Auslandskorrespondenten' Henry Vinerts aus dem Silicon Valley wurde wie so oft von Thomas Beierlein übersetzt.*

*An dieser Stelle möchte ich bei Thomas Beierlein herzlich für die zuverlässige, kontinuierliche Arbeit bedanken.*

MB

Lieber Martin,

in Deiner letzten Post sagtest Du mir, daß Dein Sohn einen Volvo mit 200.000 km auf dem Tacho gekauft hat. Mein Gott, das ist ja gerade mal ein Teenager von Auto! Mein ältester Volvo (1966) wurde über 350.000 Meilen gefahren, das sind so um die 560.000 km; der nächste (1971) kam auf ungefähr 390.000 km, und der 'neue', der 1984er GL, mit Automatikgetriebe und Klimaanlage, ist der Wagen meiner Frau mit nur 290.000 km. Mein Schwiegersohn (in Köln) sagte mir, daß keiner meiner Volvos auf den deutschen Autobahnen fahren dürfte, aber das ist eine andere Sache. Ja, VolvoVid Steht für das Auto und für die ersten drei Buchstaben meines lettischen Vornamens, aber Du hast natürlich recht: "rollendes Aussehen" ist eine andere mögliche Übersetzung.

Habe ich Dir jemals erzählt, daß es an den Volvos liegt, daß ich in die Welt von Forth gelangt bin? Die gesamte Geschichte steht in meinem Brief an die Forth Dimensions vom Januar/Februar 1993, "Volvos Drove Him To Forth, etc." ("Vovlos trieben ihn zu Forth, usw.") Die hauptsächliche Verbindung besteht zu Leo Brodie's Biographie in der ersten Ausgabe von "Starting Forth," in welchem festgestellt wird, daß Leo neben anderen Dingen gern klassische Volvos fährt. Das verband mich mit dem Buch, und ich wurde seitdem der beständigste Forth Jünger überhaupt.

Nun, was das SVFIG Treffen am letzten Samstag, dem 19. Mai, betrifft habe ich das meiste schon in meiner kurzen Notiz an Dr. Behringer behandelt, aber ich werde den Bericht für die VD hier noch einmal bringen.

Die Zuhörerschaft war auf dem letzten Treffen geringer als üblich (meist nur 15 oder 16 im Raum). Dr. Ting füllte die beiden Stunden am Morgen mit einer umfassenden Beschreibung der 32-Bit Version von eForth, welches er in VHDL schreibt. Dieses soll in einem XILINX FPGA für eine eventuelle Produktion implementiert werden, für den Fall das irgendjemand mit einer Menge Geld und Vertrauen in ein Anwendungsgebiet auftaucht. Wie Du wohl gehört hast, wird das Geld in der Halbleiterindustrie zur Zeit immer knapper, viele Betriebe schränken sich ein oder fallen sogar aus dem Geschäft heraus.

Dies scheint auch die Geschichte des nächsten Sprechers zu sein; George Nicol von Silicon Composers. Er kam um uns das zu zeigen, was er in den letzten Jahren entwickelt hat und was davon immer noch relevant ist um es am heutigen Markt einzusetzen. Allerdings hat er kein Bedürfnis die Vermarktung selbst vorzunehmen. Er hat sich entschieden seinen anderen Interessen nachzugehen, insbesondere argentinisch Tango zu tanzen und zu lehren. Und ich muß sagen, auf eine gewisse Art, erinnerte mich seine freundliche und enthusiastische Art an einen Menschen, der seine Sorgen vergessen hat und einem neuen Anfang entgegen sieht.

Der letzte Sprecher, Dwight Elvey, kam mit seiner Lösung eines Puzzles, welchen mein Interesse fand. Die April-Ausgabe der 'Scientific America' Zeitschrift hatte dieses mathematisch/physikalische Problem veröffentlicht, bei welchem 15 Gewichte von einem Balken entfernt werden müssen, welcher auf zwei eng benachbarten Gelenkpunkten abgestützt wird, wobei die Gewichte hauptsächlich über die hervorspringenden Enden verteilt sind. Das Entfernen muß in einer Reihenfolge geschehen, bei der der Balken in keine Richtung abkippt (Ihr könnt die volle Beschreibung unter <http://www.sciam.com/2001/0401issue/0401puzzling.html> nachlesen.) Dwight hatte ein Programm in FPC geschrieben welches alle möglichen Anordnungen (sind es 15!?) auf ihre Gültigkeit prüft. Aber ich würde länger brauchen das Programm zu verstehen als mit einem Bleistift und drei Blatt Papier eine Lösung zu finden. Da meine Lösung nicht die gleiche ist wie in 'Scientific America' habe ich sie an Dwight zur Überprüfung geschickt.

So! Ich denke, daß Du die Teile austreichen mußt, die für die Leser der VD nicht von Interesse sind, bevor Du es zu Dr. Beierlein schickst (falls er sich immer noch treu und sorgfältig durch die Übersetzungen meines Kalifornischen Slangs kämpft). ((And he enjoy's it each time anew. TB)) Vielleicht interessiert sich Georg Beierlein auch für das Puzzle? Ich kann kaum glauben, dass es schon zwei Jahre her ist, seit wir an seinem Rätsel mit den Glühlampen knobelten.

Anyway, das ist Alles fuer Heute. Mit besten Wuenschen,

Henry

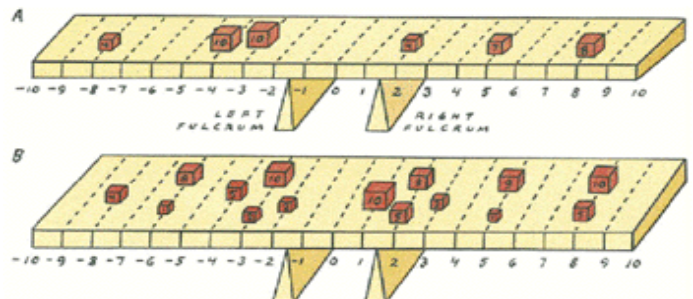


Illustration by Maris Bishofs

## Forth-Gruppen regional

**Hamburg Küstenforth**  
**Klaus Schleisiek**  
Tel. 040 / 375008-13 g  
kschleisiek@send.de  
Treffen jeden 4. Freitag im Monat  
16:30 Uhr, Fa. SEND, Stubbenhuk 10  
20459 Hamburg

**Moers Friederich Prinz**  
**Tel.: 02841-58398 (p) (Q)**  
(Bitte den Anrufbeantworter nutzen !)  
(Besucher: Bitte anmelden !)  
Treffen: (fast) jeden Samstag,  
14:00 Uhr, **MALZ, Donaustraße 1**  
**47443 Moers**

**Mannheim Thomas Prinz**  
**Tel.: 06271-2830 (p)**  
**Ewald Rieger**  
**Tel.: 06239-920 185 (p)**  
Treffen: jeden 1. Mittwoch im Monat  
**Vereinslokal Segelverein Mannheim e.V.**  
**Flugplatz Mannheim-Neustheim**

**München Jens Wilke**  
**Tel.: 089-89 76 890**  
Treffen: jeden 4. Mittwoch im  
Monat, **China Restaurant XIANG**  
**Morungerstraße 8**  
**München-Pasing**

## µP-Controller Verleih

Thomas Prinz  
Tel.: 06271-2830 (p)  
micro@forth-ev.de

## Gruppengründungen, Kontakte

**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)

Jörg Staben  
Tel.: 02173-75708 (p)

Karl Schroer  
Tel.: 02845-2 89 51 (p)

## Spezielle Fachgebiete

Arbeitsgruppe MARC4 Rafael Deliano  
Tel./Fax: 089 -841 83 17 (p)

FORTHchips Klaus Schleisiek-Kern  
(FRP 1600, RTX, Novix) Tel.: 040 -375 008 03 (g)

F-PC & TCOM, Asyst Arndt Klingelberg, Consultants  
(Meßtechnik) embedded [akg@aachen.kbbs.org](mailto:akg@aachen.kbbs.org)  
Controller (H8/5xx//, Tel.: ++32 +87 -63 09 89 pgQ  
TDS2020, TDS9092), (Fax -63 09 88)  
Fuzzy

KI, Object Oriented Forth, Ulrich Hoffmann  
Sicherheitskritische Tel.: 04351 -712 217 (p)  
Systeme (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  
[mailorder@forth-ev.de](mailto:mailorder@forth-ev.de)

Forth-Mailbox (KBBS) 0431-533 98 98 (8 N 1)  
Sysop Holger Petersen  
[hp@kbbs.org](mailto:hp@kbbs.org)  
Tel.: 0431-533 98 96 (p) bis 22:00  
Fax : 0431-533 98 97  
Helsinkistraße 52  
24109 Kiel



Möchten Sie gerne in Ihrer Umgebung eine lokale Forthgruppe gründen, oder einfach nur regelmäßige Treffen initiieren? Oder können Sie sich vorstellen, ratsuchenden Forthern zu Forth (oder anderen Themen) Hilfestellung zu leisten? Möchten Sie gerne Kontakte knüpfen, die über die VD und das jährliche Mitgliedertreffen hinausgehen?

Schreiben Sie einfach der VD - oder rufen Sie an - oder schicken Sie uns eine E-Mail !



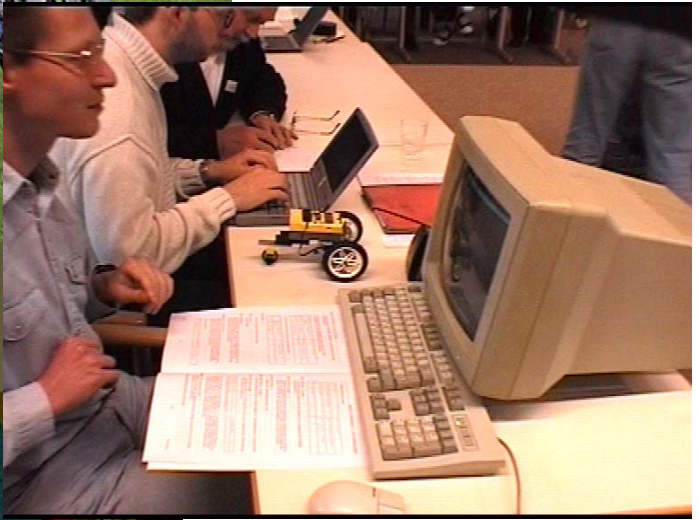
Hinweise zu den Angaben nach den Telefonnummern:

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

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



## Bilder von der Jahrestagung 2001



Gesucht: Das schönste, auflösungsgrößte dabei scharfe Gruppenbild!



In letzter Minute:

Die vierte Dimension ist schon gesetzt und teilweise gedruckt. Die Titelseiten werden als letzte Seiten fertiggestellt. Für eine Nachricht in letzter Minute ist also nur hier Platz:

Ich benutze seit Jahren den Redirector „bigfoot.de“, bisher zu meiner größten Zufriedenheit. Welchen Provider ich auch benutze – ich wechsele als bewusster Verbraucher gelegentlich – dank „bigfoot.de“ brauchten meine E-Mail-Partner sich nur eine Adresse merken; eben [mbitter@bigfoot.de](mailto:mbitter@bigfoot.de). Diese Adresse wurde auch als Redaktionsadresse im Impressum angegeben. Am 20.7. erwies es sich, dass [www.bigfoot.de](http://www.bigfoot.de) (deutsch) unerreikbaar war, statt dessen wurde zu [www.bigfoot.com](http://www.bigfoot.com) (USA) umgeleitet.

Eine ganze Weile erreichte mich Redaktionspost nicht – das Schlimme dabei: ich merkte das ja gar nicht. Wollen wir hoffen, dass es ein 'kurzer' Aussetzer bei bigfoot.de war. Letztendlich bleibt nur eines: sollte E-Mail an diese Redaktionsadresse zurückkommen (rejected), dann benutze man entweder die Adresse [VD@FORTH-EV.DE](mailto:VD@FORTH-EV.DE) oder [Martin.Bitter@t-online.de](mailto:Martin.Bitter@t-online.de). Wie lange die t-online Adresse gültig ist, weiß ich nicht, immerhin steht zum 1.8.2001 ein einschneidender Tarifwechsel an.

May the Forth be with us!  
M. Bitter