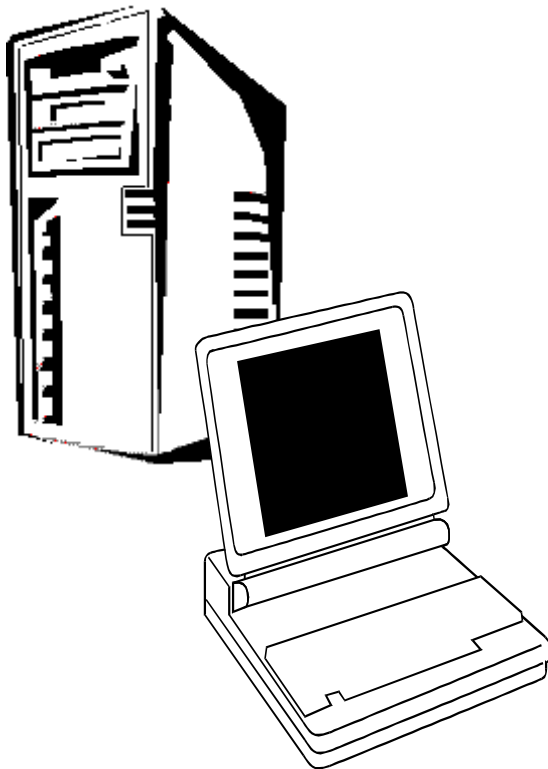
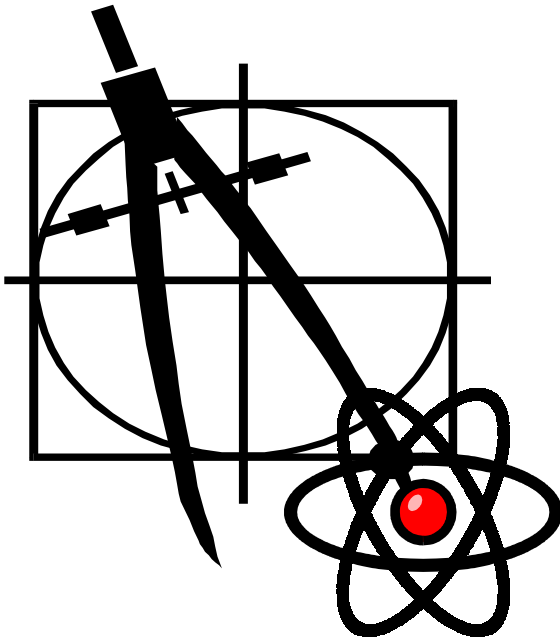


für Wissenschaft und Technik, für kommerzielle EDV,  
für MSR-Technik, für den interessierten Hobbyisten.



### In dieser Ausgabe:

#### Leserbriefe und Rezensionen

Leser schreiben, was sie interessiert

#### Neues aus der FIG Silicon Valley

Henry Vinerts berichtet

#### VD-Titelliste

Die Beiträge aller Ausgaben auf einen Blick (Teil I)

#### Gehaltvolles

Rezensionen unserer Schwesterzeitschriften

#### Die Seite für den Umsteiger

Ein Assemblieraufwurf eines Forthbegeisterten

#### Elementare Sortieralgorithmen

Ein Vergleich

#### Bericht aus Garmisch-Partenkirchen

Stimmungsbild von der FG Tagung 2002

#### Laudatio

„Inthronisierungsrede“ für einen Drachenbewahrer

#### Assemblieren als Hobby

Wo auch Forth nicht schnell genug ist...

#### Vorträge 2002

Bericht über die Vorträge in Garmisch-Partenkirchen

#### Forthweg

Mit Forth zurück zu den Wurzeln

und Vieles mehr !

## 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 SwiftX86k und seit kurzem mit Holon11 und MPE IRTC für Amtel AVR.

### **LEGO RCX-Verleih**

Seit unserem 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 Forthgesellschaft e.V., zur Verfügung stellen kann.  
Angeboten wird: Ein komplettes LEGO-RCX-Set, so wie es für ca. 230 € im Handel zu erwerben ist.

Inhalt:

1 RCX, 1 Sendeturm, 2 Motoren, 4 Sensoren und ca. 1.000 LEGO Steine.

Anfragen bitte an [Martin.Bitter@t-online.de](mailto:Martin.Bitter@t-online.de)

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

Martin Bitter

### **Dipl.-Ing. Arndt Klingelberg**

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 High-SpeedDuplicating, Tonband, (engl.) Dokumentationen und Bedienungsanleitungen.

### **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**

Joachim-Jungius-Straße 9 D-18059 Rostock  
Tel.: (0381) 405 94 72 Fax: (0381) 405 94 71

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.

<b>Impressum</b>	.....4
<b>Editorial</b>	.....4
<b>Leserbriefe</b>	.....5
Was Sie uns und den Lesern der Vierten Dimension mitteilen wollten	
<b>Neues aus der FIG Silicon Valley</b>	.....6, 28
<i>Henry Vinerts</i>	
<b>VD-Titelliste</b>	.....9
Liste aller Beiträge der bisher erschienenen VD, kategorisiert; <i>Fred Behringer</i>	
<b>Gehaltvolles</b>	.....13
Rezension der Forthwrite (FIG UK) und des Vijgeblaadje (NL); <i>Fred Behringer</i>	
<b>Die Seite für den Umsteiger</b>	.....19
Ein Assembl(i)eraufruf – Teil 1; <i>Julian Noble</i>	
<b>Elementare Sortieralgorithmen</b>	.....25
Ein Vergleich verschiedener Sortieralgorithmen; <i>Bernd Beuster</i>	
<b>Bericht aus Garmisch-Partenkirchen</b>	.....29
Über eine fröhliche Tagung berichtet <i>Friederich Prinz</i>	
<b>Laudatio</b>	.....30
Eine wirklich bemerkenswerte Lobrede auf den neuen Drachenbewahrer; <i>Martin Bitter</i>	
<b>Assemblieren als Hobby ?</b>	.....31
Ein neuer Rechner ? - da wird erst einmal mit dem Assembler getestet; <i>Bernd Beuster</i>	
<b>Vorträge 2002</b>	.....33
Eine Zusammenfassung der Tagungsvorträge; <i>Bernd Paysan</i>	
<b>Stimmungsbild</b>	.....35
Ein weiterer Tagungsbericht von; <i>Fred Behringer</i>	
<b>Bericht aus dem Drachenrat</b>	.....36
<i>Martin Bitter</i>	
<b>Weg zum Forth</b>	.....36
Mit Forth zurück zu den Wurzeln; <i>Andreas Klima</i>	
<b>Tower Power</b>	.....38
Den RCX Turm mit Strom vom PC versorgen; <i>Rolf Schöne</i>	

In der nächsten Ausgabe finden Sie voraussichtlich:

- |   |   |
|---|---|
| - die Fortsetzung der VD Titelliste,        | - Bunt in den Rubriken                                |
| - Beiträge der FG Tagung 2002,              | - die Tagungsbeiträge unserer niederländischen Gäste, |
| - den Teil 2 des Artikels von Julian Noble, | - und was immer uns unsere Leser schicken.            |

## IMPRESSUM

### Name der Zeitschrift

### Vierte Dimension

### Herausgeberin

Forth-Gesellschaft e.V.  
Postfach 16 12 04  
D-18025 Rostock  
Tel.: 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

Friederich Prinz  
Homburgerstraße 335  
47443 Moers  
Tel./Fax.: 02841-58 3 98  
E-Mail:  
VD@FORTH-EV.DE  
FRIEDERICH.PRINZ@T-ONLINE.DE

### Anzeigenverwaltung

Büro der Herausgeberin

### Redaktionsschluß

März, Juni, September, Dezember  
jeweils in der dritten Woche

### Erscheinungsweise

1 Ausgabe / Quartal

### Einzelpreis

5,11 € +Porto u. Verpackung

### Manuskripte und Rechte

Berücksichtigt werden alle eingesandten Manuskripte. Leserbriefe können ohne Rücksprache gekürzt wiedergegeben werden. Für die mit dem Namen des Verfassers gekennzeichneten Beiträge übernimmt die Redaktion lediglich die presserechtliche Verantwortung. Die in diesem Magazin veröffentlichten Beiträge sind urheberrechtlich geschützt. Übersetzung, Vervielfältigung, Nachdruck sowie Speicherung auf beliebigen Medien ist auszugsweise nur mit genauer Quellenangabe erlaubt. Die eingereichten Beiträge müssen frei von Ansprüchen Dritter sein. Veröffentlichte Programme gehen - soweit nichts anderes vermerkt ist - in die Public Domain über. Für Fehler im Text, in Schaltbildern, Aufbauskizzen u.ä., die zum Nichtfunktionieren oder eventuellem 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.



Liebe Leser,

Fünf Ausgaben lang mußte ich aus persönlichen Gründen die Arbeit an der VD aussetzen. Was meiner Planung zum Jahresbeginn 2001 unvorhersehbar den Garaus gemacht hatte, konnte die VD zum Glück nicht wirklich treffen. Martin Bitter ist eben diese fünf Ausgaben lang eingesprungen, und war Ihnen allen ein guter Editor. Sein besonderer Lohn dafür sind neben dem Dank der Gesellschaft die vielen Kontakte, die ihm die Arbeit für die VD zu Forthern in aller Welt eröffnet hat. Mehr Lohn mag er nicht. Mehr als ein Danke braucht es nicht. So sind wir Nieder-

rheiner. So sind wir Forther !

Jetzt liegt die VD wieder in meinen Händen und gleich vom Neustart weg kommt Ihnen die forthige Zeitschrift zu spät ins Haus. Das rührt vor allem daher, daß die Beiträge der Autoren für die VD immer weniger werden. Dabei ist es keineswegs so, daß Sie, die potentiellen und die tatsächlichen Autoren, nichts mehr zu schreiben hätten. Die Tagung in Garmisch-Partenkirchen hat mit ihren Beiträgen eindrucksvoll gezeigt, daß sich „unsere“ Forther mit vielen unterschiedlichen und sehr interessanten Dingen beschäftigen. Da sollte immer wieder einmal ein Artikel für die VD abfallen, ein Artikel für IHRE Zeitschrift !

Für IHRE Zeitschrift und für IHREN Verein arbeiten viele Menschen. Die meisten tun dies still und im Hintergrund. Das es sie gibt, fällt nur bei wenigen Gelegenheiten auf. Wenn auf der Jahresversammlung gleich Tagungsorte für die kommenden beiden Jahre angeboten werden, dann ist das eine solche Gelegenheit. Wenn der Drachenrat in jedem Jahr aufs neue einen würdigen Nachfolger im Amt des „Drachenbewahrers“ präsentieren kann, dann ist auch das eine solche Gelegenheit. Und natürlich zeugt auch das von einer guten Gemeinschaft, wenn Mitglieder des Vereins 1.000 km quer durch die Republik fahren, um einander zu sehen und von dem zu berichten, was sie im letzten Jahr Forthiges getan haben.

Welchen Spaß wir in Garmisch-Partenkirchen miteinander hatten, können Sie den entsprechenden Berichten in dieser Ausgabe entnehmen.

Über die Inhalte der Tagungsbeiträge berichtet wieder **Bernd Paysan**, seit Garmisch **Direktor** der Forthgesellschaft, kompetent und kurzweilig.

*Friederich Prinz*



### 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.



Hallo Martin,

ich hatte viel Schwierigkeiten mit Hacker-E-Mails. Eben habe ich mir 12 E-Mails auf meinen Computer geholt, eine Hacker-Mail "herausgeschnitten" und sehe, daß da auch eine Mail von Dir an den Stiller über Chinese Characters dabei ist. Du hörst in Kürze von mir mehr darüber.

Schön, daß Du Dir über den "Umsteiger" Gedanken machst. An sich ja gar nicht so wichtig. Aber halt doch recht interessant. Da sind wir im Deutschen ja etwas "stärker" als das ewig "kurze und klare" Englisch.

Henry meint "switchers to Forth".

Ich habe dem Alan unter anderem "language migrants" vorgeschlagen, was ihm gefiel.

Mal sehen, für was Alan sich entscheidet. Aber wenn wir die "Seite für den Umsteiger" beibehalten - und wenn Alan fortfährt, unsere VD (mit meiner Hilfe) für die Forthwrite zu besprechen - wäre es schon angebracht, wenn Alan auf lange Sicht eine brauchbare Formulierung finden könnte, ebenso prägnant wie "Umsteiger".

Fritz, vielleicht sollten wir das Problem zur Auflockerung mal in der VD ansprechen und um Mithilfe bitten. Gleichzeitig würden wir die Leser damit auf die viele Arbeit aufmerksam machen, die hinter den Kulissen läuft.

Herzlichen Gruß

Fred

*Ich denke, daß die besten Übersetzungen stets von „native Speakern“ vorgenommen werden. Mir persönlich bereiten vor allem Redewendungen und „geflügelte Worte“ immer wieder Schwierigkeiten – und zwar in beiden Richtungen. Ich erinnere in diesem Zusammenhang an die freundlichen Hinweise bezüglich VD = „veneral disease“ ;-)*

*Aus eigener Anschauung weiß ich, daß Alan seine Übersetzungen überaus gewissenhaft anfertigt. Was immer Alan vorschlägt; es wird schlicht richtig sein.*

fep

Hallo Friederich.Prinz,

hier der Bericht : (zum Microcontroller Verleih der FG; fep)

Anfragen :

Es gab keine Anfragen seit der Tagung 2001.

Weder per Post / E-Mail noch Telefonisch.

Bestand ist derzeit unverändert.

Extra Hardware :

Zusätzlich verfügbar (aber noch nicht adaptiert) sind LCD-Punktmatrixanzeigen (vermutlich 2\*20) sowie Schrittmotoren mit Leistungstreiber.

D.h., man benötigt eine selbstzufertigende Verbindung zwischen den Teilen und den Microncontrollern. Wenn ich Lust & Laune habe mach ich mich dran.

Ausblick :

Weiter gibt es ein Mitsubishi M 16C/62 Board. Der Chip darauf ist ein M 30624 FGAFP. Die Ausstattung ist 256 k Flash und 20 k RAM.

Das sollte wohl genügen für ein FORTH System.

Es gibt auch eine andere Ausführung mit CAN-Bus, den M16C/6N, die hab ich leider nicht 8-(

Thomas Prinz



M16\_62

Hallo Forther, ein kleines „Rätsel“ gefällig ?

DSL ermöglicht viele Spielereien. Musik und Film-Vorschauen (neudeutsch: Trailer) aus dem Internet herunter zu laden, gehört ebenso dazu wie Basteleien mit eigenen Servern. Ein eigener Domain-Name läßt sich ohne Kosten und ohne Aufwand von einem der vielen Dienste bekommen, die gerade dynamischen IP-Adressen (z.B. DSL Nutzern der Telekom) einen frei wählbaren Namen zuordnen. Einzige Bedingung ist hierbei natürlich, daß der gewählte Name noch nicht vergeben sein darf. DYNDNS.ORG ist ein solcher Dienst, bei dem mein Server sich mit Hilfe eines kleinen Tools (leider nicht in Forth geschrieben) (Das wäre eine echte Chance für Forth: diese Tools sind sehr gefragt !!!) immer dann anmeldet, wenn der Dienst für meinen Server eine andere IP-Adresse in der Datenbank hält, als mein Server von der Telekom zugewiesen bekommen hat. Dies ist immer dann der Fall, wenn der Server hochfährt und sich via DSL bei dem Provider Telekom neu anmeldet, und auch dann, wenn die Telekom die DSL kurzzeitig kappt, damit Leute wie ich die Flatrate auf der DSL nicht dazu nutzen, eigene Server aufzubauen, ohne teure Standleitungen zu mieten.

Wie auch immer: es funktioniert prima. WWW.FPRINZ.DYNDNS.ORG ist (meistens) ab dem frühen Morgen erreichbar und bis weit in die Nacht hinein online. Der WEB-Server ist erst im Aufbau – seit cirka einem halben Jahr –, aber der FTP-Server bietet einen Haufen „Buntes“, von dem immer wieder das Eine oder das Andere „gesaugt“ wird. Natürlich ist auch Forth darunter.

WWW.FPRINZ.DYNDNS.ORG hat aber im oberen, rechten Fenster einen Hintergrund, der ein kleines Rätsel birgt. Können Sie das lösen ?

Friederich Prinz



## Neues aus FIG Silicon Valley

Hallo, Freunde,

allen, die diese E-Mail erreicht, wünsche ich ein Frohes Weihnachten und ein Gesundes Neues Jahr. Allen Lesern -- das Beste in 2002!

Das Treffen der Silicon Valley Forth Interest Group am 15. Dezember war wieder auf mehr als eine Art unüblich, auch wenn man den Fakt außer acht läßt, daß es am dritten anstatt des vierten Samstag des Monats stattfand.

Erstens fehlte Dr. Ting, und damit auch seine Kaffeekanne.

Zweitens war die Teilnahme gering. Das mag daran gelegen haben, daß nur ein Vortrag angekündigt war und zwar über John Peter's Projekt der Erweiterung von Win32Forth, wobei er für die Hilfe Volountäre anlocken möchte (Ein Volountär ist allerdings kein „alltägliches Tier“ in diesem Teil der Welt).

Drittens war unser üblicher Klassenraum belegt. Aber die Cogswell Bibliothek war verfügbar und zu unserer Freude durften wir ihre Internet-Zugangs-Station nutzen, welche einen teuren Video-Projektor und einen großen Bildschirm besaß.

Nach ganz kurzer Zeit fand ich mich als einziger Passagier im Heck des Flugzeugs (der Bibliothek) wieder, während 10 Piloten und Navigatoren durch das Internet flogen. Ich lernte in der ersten Stunde genug, um nun zu Hause selbstständig ein wenig mit dem Hang-Gleiter zu segeln. Auf meine Anforderung fand Kevin Aperts, mit der Hilfe von Alta Vista, in nicht mehr als zwei Minuten Mitch Bradley's "Open Boot Prom" Song im Netz, aber da es in der Bibliothek war, blieben wir des Klangs beraubt. John Peters behauptete, daß [www.google.com](http://www.google.com) der schnellste Weg ist um [comp.lang.forth](http://comp.lang.forth) zu finden und konnte das auch beweisen. Durch Google landeten wir auch auf unserer eigenen Webseite, <http://www.forth.org/svfig>, auf der der Meister, Dave Jaffe, auf die Photos von unseren Forth-Tagen in 1999 und 2000 zeigte. Dort findet Ihr auch John Peter's Bericht des November SVFIG Treffens, welches ich verpaßte.

Das Photo von mir mit dem Swap-Drachen T-Shirt verursachte wahrscheinlich auch einen kurzen Besuch des deutschen Swap-Drachen im Netz, aber da die deutsche Sprache ein ziemlich seltsamer Code für unsere Forther ist, brauchte auch die neue Ausgabe der Vierten Dimension 4/2000, die ich herumgab nicht allzulange bis zu mir zurück.

Die September 2001 Ausgabe der Forthwrite, die ich von Chris Jakeman bekam, zog mehr Aufmerksamkeit auf sich.

Ein paar Worte über das "wortreiche" Win32Forth. Ich erfuhr von John Peters, daß er, wie die meisten von uns SOP's (meine Abkürzung für "Slaves of Progress"; Sklaven des Fortschritts), gezwungen war, sein Geschäftsverträge erzeugendes F-PC Programm nach Windows zu übertragen. Mit Erlaubnis von Tom Zimmer macht er seine eigenen Erweiterungen von Win32Forth. Er hat Hilfe; sogar der F-PC Guru Bob Smith war anwesend und erklärte, wie er Rechtecke zur Hervorhebung von Text für Win32Forth entworfen hat. John hatte anscheinend genug Enthusiasmus unter den Zuhörern aufgeführt - bis zu dem Punkt als Freiwillige über eine Stunde damit verbrachten den Defaultfont in WinView zu ändern (hervorgerufen durch Henrik Thurfjell's Vorschlag, die Doppelpunkte besser sichtbar zu machen). Nach dem Essen

wurde festgestellt, daß eine Verwechslung zwischen Alt- und Ctrl-Tasten das Mysterium hervorgerufen hatte.

Der Rest des Tages erlaubte die üblichen Gespräche, Tratsch, und Geplauder. Aber eine Reihe Hartgesottener verbrachte eine Stunde oder so damit, jemand anderes Code zu entziffern, der seinen Weg auf den Projektionsschirm gefunden hatte. Ich glaube es war zu dieser Zeit, als ich Kevin Appert einen seiner treffenden Kommentare ausstoßen hörte: "Such activity is not recommended for those who have a life."

Ich sollte mit dieser Bemerkung schließen. Forth lebt immer noch im Silicon Valley.

Tschuß,

*Henry*

Hallo, Fred and Chris!

die angehängte Datei enthält meinen Bericht, oder sollte ich lieber sagen, meine Erinnerungen an das SVFIG-Treffen vom 26. Januar. Da ich nicht weiß, ob Friederich oder Martin die Kopie bekommt, bitte ich Fred um eine Weiterleitung an die richtige Person.

In 12 Tagen ist schon das nächste Treffen angesetzt und ich werde versuchen unsere Mitglieder daran zu erinnern, daß es eine Menge guter Forth-Ressourcen und Leute in anderen Teilen der Welt, außerhalb des Silicon Valley, gibt. Ich würde gerne wissen, ob Chris's Einladung zum IRC, die ich weitergeleitet habe, irgendeine Reaktion hervorgebracht hat.

Nebenbei, die Einreichungsfrist für Artikel für die VD ist die 3. Märzwoche. Wie ist das Datum bei der Forthwrite?

Es interessiert Euch vielleicht, daß Huang's Artikel das Konzept separater Header von Klaus Schleisiek zitiert, und auch daß die U.S. Congress Bibliothek vermutlich CCCII -- den Chinese Character Code for Information Interchange -- angenommen hat. Weiterhin, nach der gleichen Fußnote in DDJ, kann der in England gefertigte Apricot-Computer das chinesische Dai-E Computer System mit 5000 der am häufigsten genutzten chinesischen Zeichen laufen lassen. All das gab es schon 1984! Und Forth gibt es schon ungefähr doppelt so lang. Ich wünschte, ich hätte mich besser mit all den Gelehrten hier vertraut gemacht. Ich würde sie fragen, ob ich jetzt ein chinesisches Color Forth zu sehen bekomme.

Viele Grüße,

*Henry*

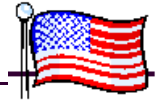
February 25, 2002

Diese Korrespondenz aus dem Silicon Valley geht an vier Empfänger gleichzeitig. Aber ich möchte ganz besonders folgendes sagen:

Vielen Dank an Martin für fünf großartige Ausgaben der Vierten Dimension, und Willkommen zurück, Friederich!

Mein Bericht vom SVFIG Treffen am 22. Februar kann in forthartiger Manier als **JSB > DLL** (in fetten Buchstaben, bitte) zusammengefasst werden. Ich vertraue darauf, daß dies auch für die kommenden Generationen gelten wird.

Es bedeutet, daß die Arbeiten von J.S. Bach großartiger sind als alle die DLL's, die von Tausenden von Code Monkeys (mangels eines besseren Wortes -- Henry) produziert wurden. Es bestärkt mich auch in meinem Glauben, daß die meisten wertvollen und dauerhaften kreativen Schöpfungen aus dem Genius und der Produktivität stammt, die in einem einzelnen Verstand enthalten sind.



In der Vormittagssitzung erfreute Dr. Ting eine Zuhörerschaft von ungefähr zehn Personen mit einer Orgel-Aufführung von Bach's "Die Kunst der Fuge". Die Orgel war ein 4,77 MHz XT in Forth programmiert und mit vier Lautsprechern ausgerüstet, die über Ting's selbstgebaute Oszillatorkarte angesteuert wurden. In diesem Paket von 1988 hat Ting nur 4 Kanäle benutzt, daher waren einige der 15 Fugen etwas abgekürzt, wobei die Musik trotzdem sehr realistisch klang. Wie Ting erklärte, liegt das daran, daß der Ton einer Orgelpfeife mit Rechteckschwingungen recht gut nachgebildet werden kann. Das ganze Konzert paßt auf vier 360 kB Floppy Disks. Und für diejenigen, die es auf ihren modernen Maschinen nochmals hören wollen, hat Ting eine 600 MB CD gebrannt, die einem, ungeachtet einiger abweichender Noten, mit ungefähr einer Stunde von Bach's schwermütiger Erhabenheit erfreut.

Ein erstaunliches Zeug! Jetzt würde ich lieber wieder meine alte Kopie von "Goedel, Escher, Bach," weiterlesen, als diesen Bericht fortzusetzen.....

Es soll genügen zu berichten, daß das Komitee, welches an Modifikationen für Win32Forth arbeiten sollte, in der Nachmittagssitzung seltsamerweise abwesend war. Eine Reihe Zuhörer verließ uns zum Essen, aber nur um durch einige andere ersetzt zu werden, die das Konzert verpaßt hatten, weil das WWW ihren Morgen beansprucht hatte. Somit wurde nichts neues zu Win32Forth gesagt oder getan.

Bob Smith, einer der ursprünglichen Mitarbeiter an F-PC, war pflichtgemäß anwesend. Aber ich denke, da Bob ein ausgebildeter Organist und Akkordeonspieler ist, war der Morgen der Hauptgrund für seine Anwesenheit. Vielleicht wird nicht allzuviel geschehen bis wir mehr über Win32Forth hören als von Bach, solange wir nicht diesen einen Verstand von Tom Zimmer aus Texas wiederfinden.

Ich regte an, daß das sechsseitige Interview von Jim Lawless mit Tom Zimmer, welches in Friederich's geschickter Übersetzung in der Vierten Dimension erschien, auch in der englischen Originalfassung durch einen Besuch der Webseiten der FIGUK erreichbar wäre.

Dave Jaffe gab noch eine kurze Beschreibung seiner neuesten Forth-Programme zur Schwellwerterfassung für Kontaktenprellung und zur Umwandlung der Ausgangsleitungen des Parallelports in einen seriellen Datenstrom. Dann war auch schon Frage-und-Antwort Stunde für die wenigen Übriggebliebenen, die nach Wegen suchten, ihre gewählten Aufgaben mit einem minimalen Zeitverlust im heutigen Informationsdschungel zu bewältigen.

Eine frohe Jagd Euch allen,

*Henry*

Hallo, Friederich,

es ist gut, dieses "Glückauf" von Dir wieder zu sehen. Ich möchte Dir ein frohes Ostern wünschen, bevor ich fortfahre und versuche, meinen jüngsten Bericht über die SVFIG zu vollenden. Ich nehme an, daß es in Ordnung ist, wenn ich gleich Kopien an Fred Behringer und Chris Jakeman sende. Falls ich das nächste mal noch jemand anderes einschließen soll, lass es mich bitte wissen.

Der Frühling ist hier und wir "putzen" immer noch

'Fenster' (Windows). Bob Smith begrüßte eine kleine Gruppe von ungefähr ein Dutzend Frühaufstehern mit seiner Zusammenfassung kürzlicher Änderungen an Tom Zimmer's Win32Forth. Da ich Jim Lawless' ursprüngliches Interview (in englisch) mit Tom nicht gelesen habe, wundere ich mich wie ich "Schmollecke" (Seite 21, VD 1/2002) wieder zurück in Tom's Worte übersetzen soll. Tom verläßt gerade diese "pouting corner" und macht mit neuen Sachen weiter. Er erlaubt dem Rest der Welt mit Win32Forth zu spielen, außer mit WinView, welches als WinEd wieder auftauchen sollte. Ich erinnere mich daran, daß Tom, als er noch im Silicon Valley war, oft Einstein zitierte, "die Dinge so einfach wie möglich, aber nicht einfacher" zu machen. Durch die von Microsoft kreierten Paradigmen ist das nicht mehr länger möglich. Es besteht keine Chance, daß Bill Gates möglicherweise aus dem gleichen Gen-Pool wie Henry David Thoreau kommt.

Tom sagte auch oft, daß Forth der Öffentlichkeit als Spiel präsentiert werden sollte, um es interessant zu machen. Jetzt scheint es mir, daß eine Anzahl von Forthern selbst, um die ganze Welt herum, Gefallen daran finden, mit Win32Forth herumzuspielen.

Tom, Du hast sie alle inspiriert! Und Bob Smith hat beim Entwurf eines Windows Icon's für den neuen WinEd Editor, vielleicht sogar ganz listig, das Markenzeichen der Einfachheit des alten Bleistiftmachers Thoreau hinzugefügt, indem er den Stengel des Feigenblattes durch einen Bleistift ergänzte!

Es war gut, Chuck Moore wieder zu sehen, allerdings als geduldigen Zuhörer und nicht als Redner auf diesem Treffen. Es ist nun fast 35 Jahre seit der Geburt von FORTH auf der IBM 1130; ich frage mich, ob Chuck möglicherweise etwas verwundert ist über all die Enkel von FORTH, die heute die Spielwiese bevölkern.

Dr. Ting hatte die Zeit festzustellen, daß Win32Forth zu kompliziert war und daß er Win-eForth für sein Projekt chinesischer Grafikzeichen verwendete. Das eForth ist von ihm und der "Win"-Teil stammt von einem jungen Mann in Taiwan.

John Peters hatte Mittagessen für alle Helfer am Projekt zur Überarbeitung von Win32Forth mitgebracht. Eine kleine Gruppe arbeitete daran bis Mike Saari mit zwei elektrischen Liegefahrrädern erschien und alle zu einer Testfahrt einlud. Das war definitiv mehr Spaß als Windows zum Absturz zu bringen, selbst für alte Forther!

Diese Fahrräder, die Mike zumeist aus Standardteilen baut, haben 21 Gänge. Ein Hybridantrieb erlaubt es zu treten oder mit einem speziellen 36-Volt Heinzmann Motor zu fahren. Mit einer Batterie sind 20 bis 40 Meilen je Ladung bei einer Geschwindigkeit bis zu 20 mph (ca. 30 km/h) möglich. Mit Unterstützung durch Treten ist eine tägliche Reichweite von 100 Meilen nicht unüblich. Ich kenne viele Teile der Welt, in der ein 1.500 Dollar Fahrrad ein Luxus wäre und ich kann mir ebenso vorstellen, daß, bei unserem ständig wachsenden Verkehrsgetümmel, es bald eine Notwendigkeit für viele Silicon Valley Pendler wird. Schaut es Euch auf den Webseiten von "ElectricBetterbikes.com" oder "SkeeterEV.com" an. Zur Zeit gibt es kein Forth an Bord des Rades, aber es mag eingebaut werden, da die Steuerung anspruchsvoller wird, sobald die Energierückgewinnung hinzugefügt wird.



## Neues aus FIG Silicon Valley

Mike's Vortrag ergab viele Fragen und ein großes Interesse bei den Zuhörern, die später auf eine Zahl von ca. 20 gefallen war. Sogar dann scheint es, daß SVFIG das Zentrum der Forth-Aktivitäten in den USA ist, soweit es den organisierten Charakter betrifft.

Bleibt dabei, Forth-Gesellschaft e.V. und -- da es über den großen Teich geht -- FIG UK ebenso!

Mit den besten Grüßen,

*Henry!*

Übersetzungen: Thomas Beierlein

*Ich danke für den Willkommensgruß, Henry, und für die vielen lobenden Worte in Deinen Briefen. Glückauf*

*Fritz*

*... weiter auf Seite 28 ...*

### – Mitarbeiter gesucht –

**Ulrich Hoffmann** möchte den Mitgliedern der Forthgesellschaft und interessierten „Dritten“ alle bisher erschienenen Ausgaben der „Vierte Dimension“ als Dateien im PDF-Format zur Verfügung stellen. Die Ausgaben der jüngeren Vergangenheit, die bereits „elektronisch“ erstellt wurden und in verschiedenen Dateiformaten vorlagen, konnten größtenteils in PDF-Dateien konvertiert werden. Zu einer größeren Zahl älterer Ausgaben bleibt aber kein anderer Weg, als die Zeitschriften seitenweise zu scannen und die dabei entstandenen „Bilder“ zu PDF-Dateien zusammenzufassen. Hierbei wird dringend Hilfe benötigt! Neben der Organisation und Koordination der anstehenden Arbeiten und neben dem Zusammenfügen der PDF-Dateien ist vor allem das Scannen sehr arbeitsintensiv und von einer Einzelperson nicht in einer akzeptablen Zeit zu leisten. Wenn SIE einen kleinen Anteil der Scan-Arbeit übernehmen wollen, setzen Sie sich bitte mit Ulrich Hoffmann in Verbindung [[Ulrich.E.Hoffmann@gmx.de](mailto:Ulrich.E.Hoffmann@gmx.de)]. Unter der angegebenen Adresse können Sie erfragen, welche Ausgaben noch gescannt werden müssen, welche Datenformate benötigt werden, welche Informationsdichte die Scans haben müssen und was immer Sie sonst an Hintergrundinformationen zu diesem Projekt der Forthgesellschaft benötigen. Bitte helfen Sie mit!

*Friederich Prinz*

*Wer war so freundlich, der Redaktion diese Nachrichten zu schicken? Einstweilen vielen Dank für die Informationen! Bitte mehr davon!*

*fep*

**Lycos** hat ein **Forthdirectory**. Unter Home > Computers > Programming > Languages > Forth > findet man einige interessante Links zu Forth. Firmen, Privatmenschen, Projekte und Organisationen. Bei den Organisationen sind insgesamt fünf FIGs aufgelistet, sowie ein 'nonprofit' Institute for Applied Forth Research (die verlinkte Seite ist von 1996, last updated: 23 September 1997) - was mir fehlt ist ein Link auf die Forth-Gesellschaft, immerhin sind wir etwas aktueller.

### **Enth**

Sean Pringle hat ein Forth-Betriebssystem für 386er und höhere PC geschrieben, das zum freien Download zur Verfügung steht.

Seine Motivation ist eine dreifache:

1. Neugier
2. Freiheit (von vorhandenen Betriebssystemen)
2. eine kongeniale Umgebung für Flux zu schaffen. Flux ist Seans Version von ColorForth.

Selbstverständlich kann man Enth auch mit 'normalem' ANSI-Forth verwenden. So ist es Forthlern beider Provinienz offen. Enth folgt in weiten Bereichen der ANSI Konvention, es wurde mit Seans eigenem SASM-Assembler geschrieben. SASM selbst ist wieder mit einem Forth verfasst worden.

Sean selbst schreibt sinngemäß:

Enth ist im Entwicklungsstadium. Es sollte als eine ALPHA Version betrachtet werden, die interessierten Menschen zum Kennenlernen zur Verfügung steht. Es ist sehr einfach damit einen Rechner zum Absturz zu bringen - deshalb werden keine Garantien gegeben. Notfalls hilft ein Kaltstart.

Wer trotz dieser Warnungen Enth kennenlernen möchte, unter <http://pringle.sphosting.com/enthrel.html> steht es Download bereit.

Sean freut sich über jedes 'höfliches' Feedback.

**Haben Sie auch „forthige“ Informationen? Bitte senden Sie diese an die Redaktion der VD!**



Technik im  
Walchensee-  
kraftwerk,  
ganz alt  
und noch  
ganz frisch!

Photo: U.Hoffmann





## Titelliste

Diese Liste wurde von Fred Behringer zusammengestellt. Sie enthält alle in unserer Zeitschrift "Vierte Dimension" jemals veröffentlichten Artikel - oder sollte sie jedenfalls enthalten. Fehlmeldungen bitte unter der E-Mail-Adresse behringe@mathematik.tu-muenchen.de . Die Übertragung nach PDF für die elektronische Version besorgte Rolf Schöne. Die Sachgruppen sind alphabetisch geordnet, die Titel innerhalb einer Sachgruppe nach dem Erscheinungsheft, innerhalb eines Heftes nach der Platzierung im Heft. Manche Artikel sind mehrfach, unter verschiedenen Sachgruppen aufgeführt. Editorials, Direktorials, Meldungen und Leserbriefe sind nur in Auswahl vertreten. Wir haben vor, diese Liste alle Jahre wieder zu aktualisieren.

<u>Thema</u>	<u>Autor</u>	<u>Heft Titel</u>
Algorithmen	Krinninger, Ch.	88-4 Rössler-Attraktor
Algorithmen	Prinz, Friederich	89-3 Der fleißige Biber
Algorithmen	Krinninger, Ch.	89-4 Graphale Pflanzen
Algorithmen	Kohl, Klaus	92-3 Fourieranalyse in FORTH
Algorithmen	Beuster, Bernd	94-1 FPFFT - Floating-Point-Fast-Fourier-Transformation (F-PC)
Algorithmen	Richter, Ulrich	95-2 Kreisalgorithmus
Algorithmen	Allinger, W.	98-4 CRC für Dummies
Algorithmen	Paysan, Bernd	99-3 Dragon Graphics (3D-Turtle-Garphics)
Algorithmen	Aguilar, Hugh	00-1 Ein Programm zum Knacken von polyalphabetischen Codes, Teil 1
Algorithmen	Aguilar, Hugh	00-2 Ein Programm zum Knacken von polyalphabetischen Codes, Teil 2
Amerika	Schleisiek, K.-P.	93-2 93-3/4, 93-5/6
Amerika	Schleisiek, K.-P.	93-3 93-7/8, 93-9/10
Amerika	Behringer, Fred	93-4 Math, Who Needs it? - Numbers - Game of Life
Amerika	Schleisiek, K.-P.	94-1 93-11/12, 94-1/2, 94-3/4
Amerika	Behringer, Fred	94-2 Interessantes aus der Forth Dimensions
Amerika	Behringer, Fred	95-2 Forth Dimensions 95-1/2
Amerika	Behringer, Fred	95-3 Forth Dimensions 95-3/4
Amerika	Behringer, Fred	95-4 Forth-Zeitschrift "Definitions"
Amerika	Behringer, Fred	96-1 Forth Dimensions 95-9/10
Amerika	Behringer, Fred	96-3 Forth Dimensions 96-3/4
Amerika	Behringer, Fred	96-4 Forth in Amerika, Rochester woanders
Amerika	Behringer, Fred	96-4 Forth Dimensions 96-9/10
Amerika	Behringer, Fred	96-4 Forth Dimensions 96-1/2
Amerika	Vinerts, Henry	96-4 From the other side of The Big Teich
Amerika	Behringer, Fred	97-1 Forth Dimensions 96-7/8
Amerika	Behringer, Fred	97-1 Forth Dimensions 96-5/6
Amerika	Vinerts, Henry	97-1 From the other side of The Big Teich
Amerika	Vinerts, Henry	97-2 From the other side of The Big Teich
Amerika	Behringer, Fred	97-2 Forth Dimensions 96-11/12
Amerika	Behringer, Fred	97-3 Forth Dimensions 97-1/2
Amerika	Behringer, Fred	97-3 Forth Dimensions 97-3/4
Amerika	Behringer, Fred	97-3 Forth Dimensions 97-5/6
Amerika	Vinerts, Henry	97-3 From the other side of The Big Teich
Amerika	Vinerts, Henry	98-1 Über die Treffen der FIG Silicon Valley
Amerika	Behringer, Fred	98-1 Forth Dimensions 97-7/8
Amerika	Behringer, Fred	98-1 Forth Dimensions 97-9/10
Amerika	Vinerts, Henry	98-2 Neues aus der FIG SV ...
Amerika	Behringer, Fred	98-2 Forth Dimensions 97-11/12
Amerika	Vinerts, Henry	98-2 Lieber Friederich
Amerika	Vinerts, Henry	98-3 Grüße!
Amerika	Behringer, Fred	98-3 Forth Dimensions 97-11/12
Amerika	Vinerts, Henry	98-4 Neues aus der FIG Silicon Valley
Amerika	Behringer, Fred	98-4 Forth Dimensions 98-3/4
Amerika	Behringer, Fred	98-4 Forth Dimensions 98-5/6
Amerika	Vinerts, Henry	99-1 Neues aus den USA
Amerika	Behringer, Fred	99-2 Forth Dimensions 98-7/8



Amerika	Behringer, Fred	99-2 Forth Dimensions 98-9/10
Amerika	Vinerts, Henry	99-2 Lieber Friederich
Amerika	Vinerts, Henry	99-3 Neues aus der FIG SV
Amerika	Behringer, Fred	99-3 Forth Dimensions 98-11/12
Amerika	Vinerts, Henry	99-4 Neues aus der FIG Silicon Valley
Amerika	Vinerts, Henry	00-1 Neues aus der FIG Silicon Valley
Amerika	Behringer, Fred	00-2 Forth Dimensions 99-1/4
Amerika	Vinerts, Henry	00-2 Neues aus der FIG Silicon Valley
Amerika	Vinerts, Henry	00-3 Neues aus der FIG Silicon Valley
Amerika	Vinerts, Henry	00-4 Neues aus der FIG Silicon Valley
Amerika	Vinerts, Henry	01-1 Neues aus der FIG Silicon Valley
Amerika	Vinerts, Henry	01-2 From the Big Teich
Amerika	Vinerts, Henry	01-3 From the Big Teich
Amerika	Vinerts, Henry	01-4 From the Big Teich
Amerika	Vinerts, Henry	02-1 From the Big Teich
ANS-Forth	Hayes, John R.	89-4 ANS-FORTH Required Words
ANS-Forth	Shifrin, Jerry	87-3 August 1987 ANS Forth Meeting Notes
ANS-Forth	Kalus, Michael	91-2 BASIS15 - ANS-Forth-Entwurf 4/91
ANS-Forth	Kalus, Michael	91-2 Neues im ANS-Forth Core Word Set
ANS-Forth	Kalus, Michael	91-3 Vergleich: ANS-Forth Core Word Set und Forth-83 Required Word Set
ANS-Forth	Bradley, Mitch	91-4 dpANS
ANS-Forth	Paysan, Bernd	93-1 ANS-Forth - der letzte Stand
ANS-Forth	Hoffmann, Ulrich	95-1 Jetzt geht's ANS FORTH !
ANS-Forth	Hoffmann, Ulrich	95-2 Jetzt geht's ANS FORTH !
ANS-Forth	Hoffmann, Ulrich	95-3 Jetzt geht's ANS FORTH !
ANS-Forth	Hoffmann, Ulrich	95-4 Jetzt geht's ANS FORTH !
ANS-Forth	Hoffmann, Ulrich	96-1 Jetzt geht's ANS FORTH !
ANS-Forth	Staben, Jörg	01-2 (ANS-Forth-)Quellcode oder (Java-)Komponente - was denn nun?
Anwendung	Kalus, Michael	86-4 Maschinenschreiben
Anwendung	Machlitt, N. Stenzel, M.	87-3 Digitalisierung von Oszillogrammen
Anwendung	Maier, D. Ecker, R. Reither, F. Ziegler, G.	88-2 Vakuum Pumpstandsteuerung für die Hochvakuumexperimentiereinrichtung (Hexe) am Beschleunigerlabor der LMU und TU-München
Anwendung	Staben, Jörg	92-2 Light-Show
Anwendung	Prinz, Friederich	92-4 Die Uhr (1)
Anwendung	Prinz, Friederich	93-1 Die Uhr (2)
Anwendung	Prinz, Friederich	93-2 Die Uhr (3)
Anwendung	Prinz, Friederich	93-1 Fis(c)hing Forth (Fischer-Technik)
Anwendung	Schemmert, W.	93-3 eForth CAN's (H8/325)
Anwendung	Staben, Jörg	93-3 Si-Nuß (Sinus auf F-PC)
Anwendung	Prinz, Friederich	93-3 Vorwärts - und dann kreuz und quer (ZF)
Anwendung	Major, Michael	
Anwendung	Kretzschmar, R.	93-4 Aufnehmer im Angebot (3 "Aufnehmer")
Anwendung	Prinz, Friederich	93-4 ZF-Recorder (automatischer Protokollierer)
Anwendung	Prinz, Ulrich	
Anwendung	Vogt, Claus	94-1 Hüftkontakt (künstliche Hüftgelenke)
Anwendung	Schemmert, W.	94-1 Der I <sup>2</sup> C-Bus - in Forth realisiert
Anwendung	Symonds, M.	94-3 Vorwärts - und dann kreuz und quer II
Anwendung	Plewe, Jörg	95-2 Forth denkt, Windows lenkt
Anwendung	Allinger, Wolfgang	
Anwendung	Bitter, Martin	96-3 Wenn alle Brunnlein fließen ...
Anwendung	Bitter, Martin	98-1 ... von der Stirne heiß rinnen muß der Schweiß?
Anwendung	Bitter, Martin	99-3 Lee-Effekt (Win32Forth)
Anwendung	Dixon, Glenn	99-4 Reed-Solomon-Fehlerkorrektur, Teil 1



Anwendung	Reilhofer, J.	00-1 Körperschallanalyse
Anwendung	Aguilar, Hugh	00-1 Ein Programm zum Knacken von polyalphabetischen Codes, Teil 1
Anwendung	Dixon, Glenn	00-1 Reed-Solomon-Fehlerkorrektur, Teil 2
Anwendung	Bitter, Martin	00-2 Farbe ins Leben (Drucken unter Win32Forth)
Anwendung	Paysan, Bernd	00-3 Ein Web-Server in Forth
Anwendung	Hoffman, Ulrich	00-4 Digitale Signaturen und eingebettete Systeme
Anwendung	Fink, Herbert	00-4 Umlaute, ein Lösungsvorschlag
Anwendung	Baden, Wil	01-1 UUENCODE und UUDECODE
Anwendung	Beuster, Bernd	01-2 Ein einfacher PostScript-Druckertreiber für bigFORTH
Anwendung	Saric, Rainer	01-2 Ende gut, alles gut (Pipettierroboter, Win32For)
Anwendung	Paysan, Bernd	01-3 MINOS Example: OpenSched GUI
Anwendung	Zobawa, Klaus	01-4 Entscheidungstabellensyntax in FORTH
Arithmetik	Illyes, Robert	86-4 A Fast High-Level Floating Point
Arithmetik	Wilton, Richard	87-2 Floating-Point Arithmetic
Arithmetik	Soeder, Andreas	88-2 Periodenlänge von Dezimalbrüchen
Arithmetik	Raschke, Frank Stüss, Frank	89-1 Floating-Point-Einbindung fürs IBM-volksFORTH
Arithmetik	Berlev, Finn	91-1 Dezimalzahlen durch Brüche angenähert
Arithmetik	Hohl, Heinrich	92-2 Festkommapaket
Arithmetik	Deliano, Rafael	97-2 Nucleus für Controller: Arithmetik
Arithmetik	Deliano, Rafael	97-3 Nucleus für Controller: Arithmetik (Forts.)
Arithmetik	Behringer, Fred	98-1 Zen Floating Point von C.H. Ting
Arithmetik	Bitter, Martin	98-2 ETT (ED) und ÖMMES (zu Tings ZFP)
Arithmetik	Behringer, Fred	98-3 Verbesserung von FDUMP und F. aus ZFP
Arithmetik	Behringer, Fred	00-3 GGT ohne Division für ZF und Turbo-Forth in 32-Bit Breite
Arithmetik	Nemtsev, N.	00-4 Leserbrief zum "Größten gemeinsamen Teiler"
Arithmetik	Bitter, Martin	01-1 Neulich am Moerser Stammtisch ...(Russische Multiplikation)
Arithmetik	Pöppe, Christoph	01-2 Rechnen mit garantierter Genauigkeit (XPA3233)
Arithmetik	Behringer, Fred	01-2 Kombinationen von k Einsen in einem n-Bit-Wort in High-Level-Forth
Arithmetik	Bitter, Martin	01-2 Karatsuba, Teil 1
Arithmetik	Behringer, Fred	01-3 Lego-Roboter und arithmetisierte Logik in Forth
Arithmetik	Pöppe, Christoph	01-3 Rechnen mit garantierter Genauigkeit (korrigierte Version)
Assembler	Sundermann, M.	89-3 Assembler im Vergleich
Assembler	Beuster, Bernd	96-3 32-Bit 386-Präfix-Assembler mit AT&T Syntax
Assembler	Behringer, Fred	99-4 BEGIN-UNTIL über 32K ab 80386 im ZF-Assembler
Assembler	Behringer, Fred	99-4 Eakers CASE für Assembler in ZF und Turbo-Forth
Assembler	Behringer, Fred	00-2 DO-LOOP über 32K ab 80386 im ZF-Assembler und Turbo-Forth
Assembler	Behringer, Fred	01-1 ASM2COM über Turbo-Forth: Warum meldet sich der RCX nicht?
Assembler	Behringer, Fred	01-3 Forth öffnet die Türen (für Umsteiger)
Assembler	Allinger, W.	01-4 MickerForth oder was tue ich mir eigentlich an? MACRO4th.asm !
Assembler	Behringer, Fred	01-4 CODE-Definitionen ohne CODE und END-CODE
Assembler	Bitter, Martin	01-4 Assembler "brute force" für den Lego-RCX
Assembler	Behringer, Fred	02-1 FINDRAMD.COM - Assemblerprogrammierung in Forth
Ausland	Behringer, Fred	96-1 Forth International (Schweden: 32-Bit-Forth)
Ausland	Behringer, Fred	96-2 Forth International (Antwort aus England)
Ausland	Behringer, Fred	98-3 Rußland, Frankreich, England und Amerika
Ausland	Behringer, Fred	99-4 Forth in Rußland
Ausland	Behringer, Fred	99-4 Forth in Taiwan
Ausland	Behringer, Fred	01-3 Forth in Taiwan, Amerika und England
Berichte	Mann, Norfried Weineck, Dietrich	87-1 volksFORTH beim Sechstagerennen in Bremen
Berichte	Stout, Graham	87-3 Das MARC4-Projekt
Berichte	Krinninger, Ch.	88-4 Auswertung der Umfrageaktion



Berichte	Prinz, Friederich	89-2 Nachrichten aus der Provinz (Moers)
Berichte	Plewe, Jörg	90-1 Bericht der lokalen Gruppe Rhein-Ruhr
Berichte	Kabitzsch, K.	90-1 Low-Cost-Automatisierung mit FORTH (aus der DDR)
Berichte	Prinz, Friederich	90-3 Brief aus der Provinz (Gruppe Moers)
Berichte	Schnitter, Heinz Teich, Johannes	90-3 FORTH-Gesellschaft e.V. intern
Berichte	Prinz, Friederich	90-4 Brief aus der Provinz (Gruppe Moers)
Berichte	Plewe, Jörg	90-4 Lokale Gruppe Rhein-Ruhr
Berichte	Schnitter, Ulrike	90-4 FORTH-Gesellschaft e.V. intern
Berichte	Prinz, Friederich	90-4 Wie Einsteiger FORTH und die FG sehen
Berichte	Klingelberg, A.	92-4 BITs vom BUS
Berichte	Schemmert, W.	93-4 ONLY Forth, ALSO andere!
Berichte	Steffenhagen, B.	93-4 Fuzzy und Forth
Berichte	Deliano, Rafael	94-2 d.c.l.f. (aus der Mailbox)
Berichte	Vogt, Claus	95-4 VD quo vadis?
Berichte	Kalus, Michael	96-2 Der Drachenrat und die Forthgesellschaft
Berichte	Prinz, Friederich	97-2 Drachenrat in Ludwigshafen
Berichte	Vogt, Klaus	97-2 Forth-Magazin VD - Quo Vadis?
Berichte	Moore, Chuck	97-3 Rede vom 26.7.97 auf der SVFIG
Berichte	Hoffmann, Ulrich	98-3 Drachenrat 1998
Berichte	Prinz, Friederich	00-2 Windows 2000
Berichte	Staben, Jörg	00-3 Alles ist Forth (Win32Forth)
Berichte	Jakeman, Chris	00-4 Mehr zur Forth Interest Group in Amerika
Berichte	Bitter, Martin	00-4 Bericht von der Media-Night in Vaterstetten
Berichte	Behringer, Fred	00-4 Bei den Forth-Robotern glänzten ihre Augen
Berichte	Paul, Ulrich	01-2 CAMs und Forth
Besprechung	Hoffmann, Ulrich	89-1 Ron Geere, FORTH: The NEXT Step
Besprechung	Behringer, Fred	98-3 "embedded" - Lesenswertes
Besprechung	Freitag, Robert	99-3 Embedded 3 erschienen
Bildschirm	Reilhofer, Joh.	88-4 Erzeugung von Bildschirmmasken
Bildschirm	Kühnel, Klaus	88-4 FORTH-WINDOWS - Fenstertechnik für die Schneider-CPCs
Briefe	Prinz, Friederich	92-2 Kein Prinz-Forth !!
Briefe	Gruppe Moers	92-2 Brief aus der Provinz (Gruppe Moers)
Briefe	Neumann, Helge	92-2 Die Forth-Gruppe Stuttgart
Briefe	Plewe, Jörg	92-2 Prügel für das Schwein in der Kiste
Briefe	Plewe, Jörg	92-3 Zuckerbrot und Peitsche
Briefe	Prinz, Friederich	92-3 Brief aus der Provinz (Gruppe Moers)
Briefe	Staben, Jörg	92-3 Allerlei von Staben
Briefe	Freitag, Robert	92-3 Boshafte Betrachtung (Keine Doku bei F83)
Briefe	Limbach, Gerd	92-4 Forth-Wiedereinsteiger
Briefe	Staben, Jörg	92-4 Das F-PC und seine Stiefbrüder
Briefe	Prinz, Friederich	93-1 Ein Streifzug durch die VD
Briefe	Kalus, Michael	93-1 ANS-Forth kommt - Interesse geht?
Briefe	Bradley, Mitch	93-1 Mögliche Wirkungen des ANS-Forth
Briefe	Klingelberg, A.	93-3 "Die Uhr" - kein TSR, ein TWC!
Briefe	Schröder, M.	93-3 Noch was zur "Uhr"
Briefe	Freitag, Robert	93-3 Hilfsbereit (Einsteigerbuch?)
Briefe	Kochenburger, A.	93-3 Anspruchsvoll (Berichte aus dem Ausland?)
Briefe	Klingelberg, A.	93-4 Die negativ-unendlichen Tatsachen des Bodens
Briefe	Prinz, Friederich	93-4 Brief aus der Provinz (Gruppe Moers)
Briefe	Deliano, Rafael	94-1 Unzufrieden
Briefe	Prinz, Friederich	94-2 Brief aus der Provinz (Gruppe Moers)
Briefe	Prinz, Friederich	94-3 Brief aus der Provinz (Gruppe Moers)
Briefe	Prinz, Friederich	95-2 Brief aus der Provinz (Gruppe Moers)
Briefe	Prinz, Friederich	95-3 Brief aus der Provinz (Gruppe Moers)



Briefe	Hendrix, Marcel	95-4 eForth for Linux
Briefe	Wejgaard, Wolf	95-4 Holon-Lite
Briefe	Prinz, Friederich	95-4 Brief aus der Provinz (Gruppe Moers)
Briefe	Hoffmann, Ulrich	98-1 Ein Leserbrief zum Doppelcolon
Briefe	Kalus, Michael	00-3 Forth-Gesellschaft 2000
Briefe	Die Redaktion	00-4 Zuschriften auf "Spaß mit Forth"
Briefe	Nemtsev, N.	00-4 Leserbrief zum "Größten gemeinsamen Teiler"
Briefe	Behringer, Fred	00-4 Zum Forth-Treuebekenntnis von Michael Kalus
Briefe	Paul, Ulrich	01-1 Zu "Atomzeit und PC"
Briefe	Güttler, Peter	01-1 Zum "RCX-Selbstbau"
Briefe	Paul, Ulrich	01-2 Forth oder nicht Forth?
Briefe	Paul, Ulrich	01-2 Leserbrief zu REORDER
Briefe	Behringer, Fred	01-2 Nachruf auf den Tod von C.E.Shannon
CASE	Staben, Jörg	89-2 Behandlung einer CASE-Situation, Teil I
CASE	Staben, Jörg	89-3 Behandlung einer CASE-Situation, Teil 2
CASE	Haase, H.-J.	95-1 Zugehörig oder nicht?
CASE	Behringer, Fred	96-4 Über Eakers CASE in Forth und das IF-Konstrukt in OCCAM2
CASE	Behringer, Fred	99-4 Eakers CASE für Assembler in ZF und Turbo-Forth
Chatten	Behringer, Fred	99-1 Erste Schritte auf einer englischen IRC-Sitzung
Compiler	Kalus, Michael	86-1 Den Compiler steuern
Compiler	Neuthe, Ralf	90-2 CrossCompiler für FORTH
Compiler	Paysan, Bernd	91-3 Ein optimierender FORTH-Compiler
Compiler	Dyja, Holger	93-4 68HC11 - noch mehr Forth - 2. Teil
Compiler	Schemmert, W.	93-4 ONLY Forth, ALSO andere!
Compiler	Paysan, Bernd	95-3 FORTH optimieren
Compiler	Deliano, Rafael	96-2 STOIC - Stack Oriented Interactive Compiler
Controller	Deliano, Rafael	97-1 Nucleus für Controller
Controller	Deliano, Rafael	97-2 Nucleus für Controller: Arithmetik
Controller	Deliano, Rafael	97-3 Nucleus für Controller: Arithmetik (Forts.)
Control-Structur	Plewe, Jörg	93-3 UCS? (Universal Control Structure)

Die Titelliste der VD wird in den kommenden VD in weiteren Teilen vollständig abgedruckt werden. Gleichzeitig können Sie die vollständige Liste mit dem Arbeitsstand März 2002 auch in Dateiform (DOC und/oder PDF) anfordern.

fep

## Gehaltvolles

zusammengestellt und übertragen  
von Fred Behringer

### FORTHWRITE der FIG UK, Großbritannien

#### Nr. 109 November 2000

Chris Jakeman begrüßt in seinem Editorial Roelf Toxopeus aus Utrecht/Holland als neues Mitglied und berichtet, daß die

Forthwrite-Download-Seite nun schon 685 Besucher gezählt hat. Keith Matthews hat seinen Posten als Schatzmeister (Kassierer) an Neville Joseph übergeben.

#### 3 Forth News

Dave Abrahams <d.j.abrahams@cwcom.net>

FIG UK - Forth-gebruikersgroep - Embedded Forth - Wil Badens Homepage - OOF-Bibliothek von Anton Ertl - pbForth - StrongForth - Ulrich Hoffmann hilft international aus mit MD5 - C.H. Ting und eine VHDL-Version des P16 - Jef Raskin und "The Humane Interface".



Holländisch 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 10 € 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 10 € 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.

### 5 BLT is not a sandwich

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

Der Autor, maßgebender Mitgestalter des internationalen Projektes "WebForth", untersucht die hervorragenden Möglichkeiten, die Blocktransfer-Einrichtungen unter WIN32For zu nutzen, um Windows "anzubohren". Bitweises Transferieren eines ganzen Blocks mit 256 Arten der gleichzeitigen logischen Verarbeitung "unterwegs".

### 10 F11-UK

FIG-UK-Hardware-Projekt. 47 engl. Pfund, Porto und Verpackung 4 Pfund (auf der Insel nur 2 Pfund), Pygmy-HC11-Forth-Registrierung 25 US-Dollar. Pygmy-HC11-Forth-Compiler mit allen Quelltexten. Über 100 Seiten Dokumentation. HC11, 8 MHz. 103x100 mm. Mehr bei: [jeremy.fowell@btinternet.com](mailto:jeremy.fowell@btinternet.com) .

### 11 Permutation by Transposition Sequence

Wil Baden <[neilbawd@earthlink.net](mailto:neilbawd@earthlink.net)>

Der Autor, bekannter Forth-Fachmann und Verfasser vieler interessanter Artikel für die Forth Dimensions, entdeckt die Forthwrite für Publikationszwecke. (Wir hatten ihn in Übersetzung auch in der VD. Vielleicht sollte man ihm auch bei uns verstärkt Veröffentlichungsgelegenheit geben?)

Alle Permutationen der Ziffern von 1 bis n niederschreiben. Ab n=9 wird es unhandlich viel. Er greift auf ein frühes Programm von ihm zurück, das in ALGOL 60 verfaßt war, und "übersetzt" es nach Forth. Verwendet wird ein GOTO in Forth, in bezug auf dessen Ausformulierung er auf seine Webseite verweist. Nach welchem Muster wurde eigentlich Wils Name in der E-Mail-Adresse permutiert?

### 16 Simple Forth Permutations

Chris Jakeman <[cjakeman@bigfoot.com](mailto:cjakeman@bigfoot.com)>

Chris macht auf eine kürzere Forth-Prozedur aus dem Archiv der Forthwrite aufmerksam. Wirklich gut, ein solches Archiv! Die Fassung stammt von Ed Hersom. Hier ist es nicht GOTO, sondern ROLL, dessen Rolle in Forth diskutabel ist.

### 18 euroForth 2000

Die diesmalige Tagung wurde von der FIG UK ausgetragen und fand vom 17. bis 19. November in Prestbury/Cheshire statt. Die Konferenzleitung lag in Händen von Malcolm Bugler, den Programmvorsitz hatte Dr. Peter Knaggs.

(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:

[www.users.zetnet.co.uk/aborigine/Forth.htm](http://www.users.zetnet.co.uk/aborigine/Forth.htm)

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)



## 20 Did you Know? - Open Firmware

Chris Jakeman <cjakeman@bigfoot.com>

Chris gibt einen erläuternden Beitrag von Elizabeth Rather aus der comp.lang.forth wieder. SUN, Apple, IBM, Motorola. Endlich weiß ich, warum bei Fehlern auf den SUN-Workstations Forth zum Vorschein kam. Ein kompakter Forth-Kern steckt in der Boot-up-Maschinerie und übersetzt vor-compilierte Treiber, um sie in Plug-and-Play-Manier an die Peripherie der jeweiligen Maschine anzupassen. Dann wird vom Forth-Kern (im PROM) das eigentliche Betriebssystem geladen. Unterbricht man den Bootprozeß durch Drücken einer bestimmten Tastenkombination, dann kann man das zugrundeliegende Forth-System zur schnellen System-Diagnose verwenden.

## 21 AGM Report

Chris Jakeman <cjakeman@bigfoot.com>

Bericht über das Jahrestreffen (Annual General Meeting). Chris Hainsworth (Vorsitzender) dankt Keith Matthews für seine bisherige wertvolle Arbeit als Finanzverwalter (Treasurer). Jeremy Fowell wurde in den Vorstand aufgenommen (sozusagen als "Minister ohne Ressort"). Das F11-Projekt blüht, WebForth schlummert momentan. Die Finanzen sind in Ordnung Die Mitgliederzahl steigt langsam wieder. Einige FIG-US-Mitglieder haben zu FIG-UK gewechselt. Ehemalige Autoren der Forth Dimensions veröffentlichen jetzt in der Forthwrite. IRC-Chatting funktioniert mit starker internationaler Beteiligung glänzend, die elektronische Forthwrite-Seite wird viel besucht, die Bibliothek hat großen Zuspruch.

## 23 Forth in the UK

Chris Jakeman <cjakeman@bigfoot.com>

Ein Artikel über Forth und die Forth-Aktivitäten der FIG UK, der im Dezember 2000 in der SIGPLAN Notices (Special Interest Group for Programming Languages), einer Schriftenreihe der ACM (der US Association for Computing Machinery) erscheint (erschien).

Was ist Forth? - Modularisierung - interaktive Entwicklung - erweiterbarer Compiler - Meilensteine - modernes Forth - Forth in UK - FIG UK - Ausblick; 14 Quellenangaben.

## 28 Vierte Dimension 3/00

Alan Wenham (Forthwrite-Award-Preisträger 2000) faßt wieder einmal glänzend zusammen.

## 31 Deutsche Forth-Gesellschaft

Diesmal wieder unsere Anzeige zur Mitgliederwerbung. Das nächste Mal sind dann wieder unsere holländischen Forth-Nachbarn an der Reihe.

## 32 Letters

5 Briefe an den langjährigen und verdienten Redakteur der Forthwrite, Chris Jakeman. Klaus Zobawa (Forth-Gesellschaft)

sucht frühere Artikel aus der Forthwrite und bekommt sie prompt, John Tasgal (FIG UK) lobt Bernd Paysan und dessen Forth-Server-Artikel, Jim Lawless (FIG US) lobt die #FIGUK-Chat-Sitzung, Norman Ngige (USA) fragt nach über BBL und ABUNDANCE, Ralph Hempel (Kanada) macht auf sein neues Buch über Forth und die Roboter aufmerksam.

## FORTHWRITE der FIG UK, Großbritannien

Nr. 113 September 2001

Chris Jakeman begrüßt in seinem Editorial zwei neue Mitglieder, einer aus USA. Er macht auf das verbesserte Aussehen der FIGUK-Homepage aufmerksam: [www.fig-uk.org](http://www.fig-uk.org).

## 2 Forth News

Dave Abrahams <d.j.abrahams@cwcom.net>

PPP.com jetzt bei [www.inventio.co.uk](http://www.inventio.co.uk) (Internet-Protokolle debuggen); <http://forthprimer.siteaddr.com>: 84 Seiten ANS-Forth-Dokumentation; Quantenmechanik unterrichten in Forth; CamelForth jetzt ANS-gebunden; neues Win32Forth mit Verbesserungen von Bernd Paysan; ppForth für Palm Pilot; Tiny Open Firmware; Color Forth; Ficl 3.00; iForth-Handbuch und -Wortliste; VFX Forth

## 6 Treating Data as Source

Jennifer Brien <jennybrien@bmallard.swintemet.co.uk>

Forth enthält eine Reihe von nützlichen Eigenschaften, mit deren Hilfe man Quelltexte interpretieren kann. Leider hilft einem das nicht sehr viel weiter, wenn diese "Quelltexte" aus anderen Quellen als aus Forth stammen. Das ist der erste Teil eines für die euroForth 2001 vorgesehenen Vortrags, der das Problem über eine ANS-Forth-Modifikation angeht und als einfaches Beispiel einen (auf eine Methode von Bernd Paysan zurückgehenden) XML-Parser angibt. Der zweite Teil erscheint in der nächsten Ausgabe der Forthwrite.

## 10 From the 'Net - 4-bit Forth

Tom Zimmer

Tom hatte, wie er berichtet, vor zehn Jahren bei Samsung die Aufgabe, für 4-Bit-Prozessoren ein Entwicklungssystem zu entwickeln - in Forth. Er ist heute noch "stolz darauf".

## 12 F11-UK

jeremy.fowell@btinternet.com

Und wieder die Daten des UK-Hardware-Projekts: Wirts-PC unter Windows oder DOS. Kein Adapter benötigt. Multi-Tasking-Pygmy-Forth. Inline-Austesten der Anwendungen.



### 13 euroForth 2001

Und noch einmal ein Aufruf: 26. November 2001 auf Schloß Dagstuhl bei Saarbrücken. 2 Tage, 350 C=. Chuck Moore ist als Ehrengast angekündigt. Wenn diese Meldung erscheint, wird alles schon gelaufen sein.

### 14 FIG UK - AGM

Die nächste Jahresversammlung (Annual General Meeting) findet (fand) am 26. Oktober bei Doug Neale ("Forth-Büro") statt. Neville Joseph, der Schatzmeister, veröffentlicht auf Seite 15 einen Rechenschaftsbericht.

### 15 Revenue Account

**Neville Joseph <naj@najoseph.demon.co.uk>**

Die Mitgliederzahl ist angestiegen. Der Finanzrückhalt beträgt fast doppelt soviel wie im Vorjahr. Fast die Hälfte der Einnahmen geht für Druckkosten für die Forthwrite drauf. Fast 4/10 der Druckkosten kommen für Porto hinzu.

### 16 A Call to Assembly 1/3

**Julian Noble <jvn@virginia.edu>**

Der Autor, wohlbekannt durch sein Buch "Scientific Forth", setzt sich glühend für eine Verwendung von CODE-Definitionen dort ein, wo sie angebracht sind. Forth als flexibles Assemblierwerkzeug. In diesem ersten Teil spricht er ausführlich über die Umkehrung der Bitanordnung in einem Computerwort (z.B. 10111011 --> 11011101). Er benötigt das für die Schnelle Fouriertransformation (FFT). (Er ist Professor am Institut für Kern- und Teilchen-Physik der Universität von Virginia.) Im Forth-Assembler, er nimmt F-PC und die Intel-Prozessoren als Beispiel, sind solche Aufgaben mit Leichtigkeit zu bewältigen. Anmerkung des Rezensenten: Einbettung in die umgebende Hochsprache, hier also Forth, ist eine derartige Selbstverständlichkeit, daß man sich scheut, zur Kenntnis zu nehmen, daß andere Sprachen mit dem hierfür eigens erfundenen Begriff des "Inline-Codes" Schwierigkeiten haben oder zumindest hatten.

### 24 Vierte Dimension 2/01

**Alan Wenham <101745.3615@compuserve.com>**

Alan bespricht den Inhalt. Er macht die Leser der Forthwrite ausdrücklich darauf aufmerksam, daß sie Artikel daraus über ihn in Kopie beziehen können.

### 27 "Quikwriter" Project Launch

Es ging um die von Jenny Brien propagierte Aufgabe, ein Einhand-Text-Eingabewerkzeug für Behinderte zu entwickeln. Das Projekt wurde jetzt offiziell zu einem solchen erklärt. Es soll auf das F11-UK-Projekt aufsetzen. Jeremy Fowell übernimmt die Organisation. Zuschriften an die von Graeme Dunbar eingerichtete Mailing-Liste.

### 28 Dutch Forth User Group

Diesmal wieder die Werbeanzeige unserer holländischen Forth-Nachbarn.

### 29 Did you Know? - smart cards

**Chris Jakeman <cjakeman@bigfoot.com>**

Kreditkarten-Programmierung kann in Forth sehr günstig gestaltet werden. Forth Inc. und eine australische Firma haben sich darin versucht. Die Kunden wurden jedoch vom Wort "Forth" abgeschreckt. Sun pumpt Riesengelder in Propaganda für die Kartenprogrammierung in Java. Mit Erfolg. Die Karten allerdings werden dadurch wesentlich teurer.

### 30 Win32Forth Fonts

**Dave Pochin <davep@sunterr.demon.co.uk>**

Die Liste ".fonts" in Win32Forth enthält viele Parameter. Diese lassen sich verändern. Dave diskutiert die Möglichkeiten. Der Teufel steckt im Detail. Was Windows nicht mag, ersetzt es durch Ähnliches. Dies und alles andere von Dave findet man auch unter <http://www.sunterr.demon.co.uk/>.

## FORTHWRITE der FIG UK, Großbritannien

### Nr. 115, Januar 2002

Chris Jakeman begrüßt in seinem Editorial den für die Forthwrite ersten Bericht von Henry Vinerts über die FIG Silicon Valley. Er dankt der Forth-Gesellschaft für die Genehmigung, die ursprünglich für die FG geschriebenen Berichte in Originalfassung zu übernehmen. Außerdem macht er auf zwei kommende Ereignisse aufmerksam: euroForth 2002 und Forth-Tagung ("German FIG").

### 2 Forth News

euroForth 2002 im September in Wien; FICL 3.02; CGI-Skripts unter Apache für Windows; F11-UK-Mailing-Liste; Hans Bezemers 4th-Compiler unter anderer Adresse; über 30 Forth-Quellen auf Neil Bawds Homepage.

### 3 The End of the Line

**Dave Pochin <davep@sunterr.demon.co.uk>**

Dave, den ich per E-Mail-Verbindung von unserer internationalen gemeinsamen Arbeit an WebForth her kenne, hat sich drei Jahre lang damit beschäftigt, "das Ungeheuer Win32Forth zu zähmen", wie Chris Jakeman sich ausdrückt. Hier ist der letzte Artikel aus Daves Serie. Mehr findet man auf seiner Website <http://www.sunterr.demon.co.uk/>. Im Artikel geht es um SetTextAlign.





## 6 From the 'Net - a non-English view Michael Gassanenko

Über den Sinn oder Unsinn mancher (natürlich (?) englischer) Forth-Bezeichnungen und den Klang, den sie in den Ohren eines Menschen anderer Muttersprache haben (könnten). Eine Antwort aus russischer Sicht auf eine Frage von Marcel Hendrix in comp.lang.forth. Der Rezensent: Und was ist mit jenen, die überhaupt keine andere Sprache verstehen? Seien wir froh, daß wir Forth haben, wo schließlich jede(r) alles sofort "eindeutschen" kann, wenn ihr(m) der Sinn danach steht. Kein Mensch denkt sich etwas bei "FIG", und über "tag" und "Tag" wundert sich keiner. Und "VD" durch "4D" zu ersetzen (wir kennen alle die Diskussionen in der FG darüber), ist sicher übertrieben und eine Verbeugung in nur einer einzigen Richtung. (Es gibt ja auch andere Sprachen.) Michael liefert hier einen wertvollen Beitrag und einen schönen Anlaß, wieder einmal über sprachpsychologische Empfindsamkeiten nachzudenken - hier gleich aus internationaler Sicht, wie es sich heutzutage gehört.

## 8 The Semantic Web Chris Jakeman <cjakeman@bigfoot.com>

Chris gibt ein paar Ansichten von Tim Berners-Lee - der 1989 das Web erfunden hat - zu diesem Thema wieder. HTML liefert Formatierungsinformationen, XML liefert Strukturinformationen. Was das Web braucht, ist Kontextbezogenheit. RDF könnte die Antwort sein. FAQs hierzu findet man unter <http://www.w3.org/RDF/FAQ>.

## 10 F11-UK jeremy.fowell@btinternet.com

"Alles, was Sie brauchen, finden Sie auf dieser preiswerten Forth-Controller-Karte in Industriequalität: PygmyForth, Multitasking, Assembler, ... Sofort lieferbar."

## 11 JenX Re-visited - A Simple XML Parser Jenny Brien <jennybrien@bmallard.swinternet.co.uk>

Ein Vortrag, den Jenny Brien auf der euroFORTH-Tagung 2001 hielt, und gleichzeitig eine erweiterte Fassung eines Berichtes, der im Juli 2001 in der Forthwrite erschien. Für den Einpaß-Parser wird ein neuartiger Konstrukt, SERVANT, eingeführt. Aus dem Inhalt: "Ein Einpaß-Interpreter nimmt einen String auf führt an ihm eine Operation aus, die auf dem Inhalt des Strings beruht. Wird kein String erkannt, wird eine allgemeine Ausweichoperation eingeleitet. Bei der Operation kann es sich um eine CASE-Konstruktion handeln; wenn aber der String aus nur einem einfachen Wort besteht, ist es auch möglich, die Operationen in einer Wortliste auszuführen und SERVANT anzuwenden."

## 18 Nominations for the FIG UK Awards - 2001

Ein Aufruf, Vorschläge einzureichen: (1) für den besten Beitrag zur Entwicklung von Forth, (2) für den besten Beitrag in der Forthwrite.

## 19 A Call to Assembly 3/3 Julian Noble <jvn@virginia.edu>

Der dritte Teil der dreiteiligen Artikelserie über die Verwendung von Assembler in Forth. (Das letzte Mal habe ich, der Rezensent, im Titel leider 1/3 geschrieben, wo da doch selbstverständlich 2/3 hingehörte.) Diese Artikelserie war ursprünglich für die amerikanische Vereinszeitschrift Forth Dimensions gedacht, die ja nun leider sanft entschlafen ist. Julian richtet sein Augenmerk diesmal auf die iterative Erzeugung von sphärischen Besselfunktionen. In High-Level-Forth ist das - er präsentiert die Programmteile - gut, aber zu langsam. Forth-Assembler ist gefragt. Er nimmt Win32Forth. Auf die Arbeiten von Ting, Wenham und Behringer zur ganz leichten und schnellen Einbeziehung von FPU-Befehlen in ZF und anderen F83-Systemen geht der Autor nicht ein. Das Fragezeichen in der Normalisierungsformel ist nett. Sowas kann aber auch selbst in mathematischen Fachzeitschriften passieren. Die Auflösung steht auf Seite 24. Jedenfalls brauchen wir uns über fehlende Summenzeichen (VD 4/2001) keine Gedanken mehr zu machen. - Das sind jedoch alles unbedeutende Kleinigkeiten. Der Artikel - alle drei Artikel - ist/sind gut.

## 27 A Safer Mini-OOF Chris Jakeman <cjakeman@bigfoot.com>

"Bernd Paysans Mini-OOF ist unübertroffen und bleibt die kleinste objekt-orientierte Erweiterung von ANS-Forth", sagt der Autor, Chris Jakeman. "Mini-OOF ist wirklich minimal und enthält keinerlei Sicherheitsüberprüfung. Wenn die Methode, auf die das Objekt auf dem Stack angewandt wird, für die Klasse dieses Objektes ungeeignet ist, bricht das Forth-System in aller Regel zusammen." Chris fügt ein paar sicherheitsschaffende Zeilen Forth hinzu, die zur Ausführungszeit abgeschaltet werden können. Alles zusammen, Bernd Paysans Mini-OOF und Chris' Zusatz, wirklich nur ein paar Zeilen insgesamt, wird voll wiedergegeben.

## 31 Across the Big Teich Henry Vinerts <volvoid@aol.com>

Henrys Berichte über die Treffen der Silicon Valley Forth Interest Group erscheinen ab jetzt auch in der Forthwrite (natürlich in Originalfassung). Chris Jakeman schreibt im Vorspann: Dieses Material wurde von Henry Vinerts für die Vierte Dimension zusammengestellt und wird hier mit Genehmigung der Forth-Gesellschaft (German FIG) wiedergegeben.

## 33 German FIG Annual Conference 2002

Eine englische (vom Rezensenten übersetzte) Version der Ankündigung unserer Forth-Tagung 2002. Ein wirkliches Zeichen der Verbundenheit der englischen Forth-Freunde mit uns. Wir können Chris dankbar sein.

## 34 Dutch Forth Users Group

Diesmal wieder eine Anzeige zur Mitgliederwerbung der holländischen Forth-Freunde und zum Bezug des Vijgeblaadjes.



Die Gulden-Angabe müßte durch den Euro ersetzt werden.

### 35 Letters

Ein Brief von Boris Fennema, neu bei Forth, mit einer Frage an Chris Jakeman über OO-Forth und Datenstrukturen. Chris gibt die Frage an die Forthwrite-Autoren weiter.

### 37 Forthwrite Index

Über 400 Eintragungen der seit 1990 erschienenen Forthwrite-Artikel, nach Thema, Autor, Datum und Titel geordnet. Die neueren Eintragungen sind fettgedruckt. Es wird darauf aufmerksam gemacht, daß frühere Forthwrite-Ausgaben über den Bibliotheksdienst kostenlos entliehen werden können.

## Gehaltvolles

zusammengestellt und übertragen  
von Fred Behringer

### VIJGEBLAADJE der HCC Forth-gebruikersgroep, Niederlande

#### Nr. 29, Dezember 2001

Diese Ausgabe dreht sich um das holländische Roboter-Selbstbauprojekt und wird von dessen Wortführer, Willem Ouwerkerk, den Vorsitzenden der HCC-Forth-gebruikersgroep, allein bestritten. Schade, daß man so selten einem anderen Autor begegnet. Die Aktivitäten der Fgg scheinen enorm groß zu sein und genügend andere Mitwirkende gibt es, wahre Teamarbeit, aber schreiben tut keiner.

#### Forth op de HCC-dagen

Willem Ouwerkerk <w.ouwerkerk@kader.hobby.nl>

Willem weist auf die kommenden HCC-Tage hin (Hobby Computer Club). (Die Fgg ist eine Untergruppierung des HCCs. Mitgliedsbeiträge für die Fgg werden an den HCC abgeführt, wofür sich gewisse Anrechte auch in Hinsicht auf die anderen Untergruppierungen des HCCs ergeben.) Die Fgg wird auf den HCC-Tagen einen eigenen Stand unterhalten ("die Lötkolben werden gerade vorgewärmt").

Betaversionen von AVR-ByteForth und 8051-ByteForth können ausprobiert werden, Ushi, der Roboter, wird vorgeführt, Albert van der Horst wird sein 32-Bit-Ciforth (für DOS im Protected Mode, Linux, Windows und Stand-alone) erläutern.

#### De belofte voor 2001

Willem Ouwerkerk

In diesem fünften Teil der Beschreibung des holländischen Roboter-Bauprojekts (Ushi, die an der Tischkante entlang

fährt, ohne runterzufallen) wird in der Hauptsache ein Ultraschall-Abstandsmesser (3-200 cm) besprochen. Es wird ein gut beschriftetes Schaltbild angegeben, nach welchem ich (der Rezensent) mich ohne weiteres an einen Nachbau herantrauen würde. Außerdem gibt der Autor ein vollständiges Programm in AVR-ByteForth 1.00 an. Ein ähnlicher abstandsmessender Sensor wurde schon in Vijgeblaadje 7 beschrieben, ein Infrarot-Sensor mit dem GP2D02 in Heft 23. Der hier beschriebene Sensor auf Ultraschallbasis (40 kHz) kommt ohne teure Einzelteile von Conrad aus.

### VIJGEBLAADJE der HCC Forth-gebruikersgroep, Niederlande

#### Nr. 30, Februar 2002

Die erste Ausgabe des Vijgeblaadjes (wirklich die erste - ich kenne sie alle), in welcher weder Forth noch ein Schaltbild noch eine Prozessorbeschreibung noch irgendeine mathematische Formel auftritt. Trotzdem interessant. Ein Anfänger und ein Fortgeschrittener philosophieren über Forth.

#### Zelf programmeren, hoe doe je dat?

Ben Koehorst

Forth-Philosophie eines Eingeweihten. Die allbeliebte Zeitschrift Electuur (bei uns Elektor) entwickelt Projekte, bei denen alles "vorprogrammiert" ist. Der echte Amateur möchte selbst programmieren. Das BASIC-Programm für das Electuur-Projekt "Wobbel" benötigt 312 Bytes. Nach ByteForth für die Atmel-Familie AT89Cx051 übertragen, benötigt es nur noch 119 Bytes. Von Grund auf in Forth programmiert, schrumpft die nötige Bytezahl auf 100.

#### Forth en Robot

Ernst Kouwe

Das Roboter-Projekt Ushi aus der Sicht eines Anfängers: "Das war es genau, was ich wollte. Programmieren, löten und etwas aus eigener Kraft auf die Beine stellen, was funktioniert."

#### Ausspruch des Quartals:

**„Das Buch liegt zu weit hinten“**

Zum Ausgleich der fehlerhaften Programmierung des RCX, der den Schriftzug „Forth 2001“ auf ein Blatt Papier zeichnen sollte, hatte Martin Bitter dem RCX ein Buch zur Beschwerde aufgelegt. Das hat – selbstverständlich – auch nicht funktioniert. Merke: Programmierfehler liegen immer an unpassender Literatur !  
;-) fep



=====  
**Die Seite für den Umsteiger**  
=====

## **Ein Assembl(i)eraufruf (1/3)**

**Julian Noble** <jvn@virginia.edu>

Institut für Kern- und Teilchenphysik der Universität von Virginia, Charlottesville, VA 22901

Erster Teil eines von Fred Behringer übersetzten Artikels, der ursprünglich für die nun leider eingestellte amerikanische Zeitschrift *Forth Dimensions* vorbereitet wurde. Der Artikel erschien kürzlich in der *Forthwrite* (113-115) der englischen *Forth-Freunde*. Wir danken der *Forth Interest Group UK* für die freundliche Befürwortung der deutschen Übersetzung.

**Stichworte:** Assembler, Tutorial, Forth-Philosophie, Bitmanipulation, Umkehr der Bitreihenfolge

### **Einleitung**

Forth-Programmierer neigen dazu, den in den meisten Forth-Systemen [6] zu findenden Assembler für selbstverständlich zu halten. Wir scheuen uns aber häufig davor, Assembler-Befehle zu verwenden, da sie nicht portierbar sind; umso mehr, als in den ANS-Forth-Bestrebungen Portierbarkeit ein wichtiges Programmierziel darstellt. Zum Assembler nehmen wir nur dann Zuflucht, wenn Ausführungszeiten wesentlich werden oder wenn wir unser System auf unterster Stufe - direkter Zugriff auf Ports, Laufwerke oder Anzeigergeräte - angehen müssen.

Was tun aber die Mitglieder der Forthgemeinde in allererster Linie (wenn sie nicht gerade in Forth programmieren)? Sie versuchen, die anderen Programmierer, die immer noch mit Sprachen geringerer Qualität herumwursteln, zu Forth umzuerziehen. Schon seit langem bin ich zu der Überzeugung gelangt, daß wir bei unseren Bekehrungsversuchen eine ganze Menge verschenken, indem wir viel zu stark die Vorzüge des High-Level-Teils von Forth anpreisen, seine Erweiterbarkeit, die Abstraktionskraft, die Einfachheit, die Eleganz und so weiter und so fort. Ich denke heute, es wäre wohl besser, Forth dem ungläubig Außenstehenden über den Assembler näherzubringen.

## **Es wäre besser, Forth den ungläubig Außenstehenden über den Assembler schmackhaft zu machen.**

Zur Begründung meiner Einstellung lassen Sie mich kurz rekapitulieren, was Assembler eigentlich sind und warum sie überhaupt existieren. Computerprogramme bestehen im wesentlichen aus Zahlenfolgen, im allgemeinen Binärzahlen (Basis 2) oder Hexadezimalzahlen (Basis 16). Da das menschliche Gehirn sich nie soweit entwickelt hat, daß es Zahlen beliebiger Basen verwenden könnte, litt das Zusammenwirken von Mensch und Maschine zu Zeiten, als der Computer noch über Steckbretter und Schalter programmiert wurde, unter Geschwindigkeitsmißverhältnissen. Das Programmieren über Steckbretter und Schalter, in meiner Jugend noch üblich, war mühsam und fehleranfällig.

Zum Glück können sich die heutigen Computer auf spezielle Programme verlassen, die die für das menschliche Auge verständlichen Textdarstellungen der Befehle (mnemotechnische Kürzel) in Zahlencodes verwandeln, eben die Assembler. Gute Assembler akzeptieren auch Makrobefehle ("Pseudo-Opcodes") und Operationen, die so nützliche Hausaufgaben erledigen wie das Ansprechen von Variablen oder Konstanten oder häufig gebrauchten Befehlssequenzen über Namen statt über Adressen [7].

Aber selbst mit solchen Hilfsmitteln ist das Schreiben eines längeren Programms in Assemblersprache kein leichtes Unterfangen. Es ist schwer, Maschinenprogramme zum Funktionieren zu bringen, schwer, sie zu verstehen, und schwer, sie auf andere Maschinen zu übertragen. Zur Bereitstellung besserer Mensch-Maschine-Schnittstellen, zur Bereitstellung standardisierter Datenstrukturen und Operationen, die dem Benutzer den größten Teil seiner Arbeiten abnehmen und diese auf standardisierter Weise in Maschinensprache übersetzen, wurden Hochsprachen erfunden. Moderne optimierende Compiler können Maschinencode erzeugen, der höchstens zweimal soviel Zeit beansprucht wie der beste von pfiffiger Hackerhand erzeugte Code. Solche Entwicklungen unseres Lebens haben zu der Erklärung verführt, daß Maschinencodeprogrammierung überholt sei [8].



## Assemblieren leicht gemacht

Was passiert, wenn wir es mit Problemen zu tun bekommen, die in High-Level-Forth keine vernünftige Lösung haben? Speicherplatzbeschränkungen, verzweifelte Suche nach höherer Geschwindigkeit oder Operationen, die in Maschinenregistern ausgedrückt trivial erscheinen, auf höherer Sprachebene aber recht zeitaufwendig und umständlich sind, z.B. die Umkehr der Bitreihenfolge für die schnelle Fourier-Transformation (Fast Fourier Transformation) oder das unmittelbare Ansprechen der Ports, führen uns - wenn auch nur zögernd - dazu, unser von der Verfassung garantiertes Recht, zu assemblieren (englisches Wortspiel: auch "uns zu versammeln"), auszuüben [9].

### Unser von der Verfassung garantiertes Recht zu assemblieren

Ein Großteil der modernen Programmiersprachen erlaubt den Einbau von Maschinensprachprozeduren, die getrennt (also außerhalb des Compilerprozesses) assembliert wurden, und verbindet damit die Leichtigkeit, mit der in einer Hochsprache programmiert werden kann, mit den Vorteilen der Assemblersprache. Eine solche Mischprogrammierung ist deshalb so wertvoll, weil die meisten Programme den größten Teil ihrer Zeit mit der Ausführung von relativ wenigen Befehlen zubringen. Spaltet man derartige Engpässe in Einzelroutinen auf und programmiert diese per Hand, so kann sich das zu großem Leistungsgewinn aufschaukeln. Die Vorgehensweise ist gewöhnlich folgende:

1. Man programmiert, überprüft und korrigiert alles in der Hochsprache.
2. Mit Hilfe eines Profilers oder über algorithmische Analyse ermittle man diejenigen Teile, die gewinnbringend in Maschinensprache programmiert werden können.
3. Man unterziehe sich der ermüdenden Begleiterscheinung des Assemblierens, des Linkens und des Austestens der handcodierten Teile.

Vielfach ist jedoch Punkt 3 derart beschwerlich, daß er nur in allerletzter Verzweiflung nicht von der Verwendung auch nur des geringsten Maßes an Assemblerprogrammierung abschreckt.

Was wir eigentlich brauchen, ist eine Möglichkeit, Programmteile in Maschinensprache isoliert auszutesten, sie also als getrennte Programme laufen zu lassen. Wenn wir es nicht mehr nötig haben, speziell zum Austesten geeignet modifizierte Programmversionen zu compilieren, die Teilprogramme zu assemblieren und das Ganze zu einem ausführbaren Programm zusammenzuführen, können wir den Compiler-Test-Debug-Zyklus zu einem einzigen Vorgang zusammenziehen. Sind wir mit den Maschinencodeteilen erst einmal zufrieden, können wir diese (re)assemblieren und sie in ein oder höchstens zwei Schritten in das (compilierte) Hauptprogramm einfügen.

Forth liefert uns eine Abkürzung, die die Assemblerprogrammierung genauso einfach erscheinen läßt wie das Programmieren in einer Hochsprache. Bei Problemen, mit denen ich mich beschäftige (numerische und algebraische) setze ich gern Forth ein. Forth ist aber nicht jedermanns Sache. Außerdem verhindern häufig Vorurteile des Managements den Einsatz von Forth bei kommerziellen Produkten. Will man jedoch isolierte Maschinencodestücke ganz schnell austesten, will man sich in Vorentwürfen einen Überblick verschaffen, dann sucht Forth seinesgleichen - dann lohnt es sich, Forth einzusetzen, auch wenn man sich gezwungen sieht, das Endergebnis hinterher in C oder C++ auszudrücken.

### Forth liefert uns eine Abkürzung

Assembler, Cross-Assembler und Decompiler sind in Forth so kompakt, daß Programmierer, die an andere Sprachen gewöhnt sind, kaum glauben können, daß alles Versprochene auch wirklich geleistet wird. In dem kommerziellen Forthsystem, das ich gewöhnlich verwende, sitzt der Quelltext des herkömmlichen (postfix-notierten) Forth-Assemblers in einer etwa 14 Kilobyte langen Datei und erweitert das schon compilierte System um etwa 6 Kilobyte an compiliertem Code. Ein etwas stärker ausgearbeiteter Assembler (in einem Public-Domain-Forth), der Präfix-Notation erlaubt, umfaßt 31 Kilobyte an Quelltext und hängt an das System 8 Kilobyte an compiliertem Code an. Der Quelltext eines generischen Forth-Cross-Assemblers für den Motorola 680x0 beläuft sich auf 16 KB. Und der Assembler für Intels 80486 und den Pentium, der mit einem Windows-basierten Forth-System ausgeliefert wird, fällt mit seinen 85 KB Quelltext vergleichsweise immer noch in die Klasse der Leichtgewichte. Man vergleiche das mit dem altherwürdigen 16-Bit-Assembler MASM.EXE (v.2.0), der auch schon etwa 74 KB lang war.

Der Forth-Assembler ist in Forth geschrieben. Er arbeitet also genau so wie jede andere Folge von Forth-Worten. Zum Compilieren einer neuen Definition aus Assembler-Befehlen heraus werden analog zu `:` und `;` die Worte **CODE** und **END-CODE** verwendet. Statt aber Adressen von schon vordefinierten Worten aus dem Dictionary aneinanderzureihen, assemblieren die Assembler-Befehle ein neues Maschinencode-Fragment, das die Opcodes des Zielprozessors enthält. Ich werde die Arbeitsweise an zwei beliebigen Public-Domain-Forth-Systemen aufzeigen, nämlich an F-PC und dessen direktem Abkömmling Win32Forth, beides Ziehkinder von Tom Zimmer, die im Netz unter <http://www.taygeta.com> zur Verfügung stehen.



Jemand, der Maschinencode schreiben will, hat es bei Public-Domain-Forth-Systemen insofern leicht, als diese ihm den Zugang zu den innersten Kern-Worten gewähren. Letztere dienen ihm dann als bequeme Beispiele für die Wirkungsweise des Assemblers und zeigen ihm, wie er die einfacheren Operationen in Intel-80x86-Assembler zu programmieren hat.

In der vorliegenden Artikelserie werden drei Beispiele des Entwicklungsprozesses gegeben: **STIB** [10], eine Routine, die die Reihenfolge der Bits in einer (Binär-)Zahl umkehrt, so wie es bei der Schnellen Fouriertransformation (FFT) benötigt wird, **UCASE**, eine Routine, welche alle Kleinbuchstaben in einer ASCII-Zeichenfolge in Großbuchstaben umwandelt, wobei Ziffern und Satzzeichen so belassen werden, wie sie sind, und ein Programm zur Berechnung von sphärischen Besselfunktionen (eine rechenintensive Anwendung), bei der das zentrale Unterprogramm in Assembler programmiert wird, um das Maximale an Geschwindigkeit herauszuholen. Im weiteren Verlauf gehen wir davon aus, daß der Leser mit den Assembler-Kurzbezeichnungen der Maschinenbefehle für die Familie der Intel-80x86-Prozessoren vertraut ist. Hier und da werden wir deren Wirkungsweise etwas näher beleuchten. Dem Leser bleibt jedoch empfohlen, ein Standardwerk der Assemblerprogrammierung zu Rate zu ziehen.

## Umkehr der Bitreihenfolge

Die Routine **STIB** zur Umkehr der Bitreihenfolge kann in High-Level-Forth wie folgt geschrieben werden:

```

: STIB ( k n -- n' ) \ Umkehr der Bitreihenfolge
  0 SWAP ( -- k 0 n ) \ n' zunächst auf 0, also ( -- log2[N] n' n )
  ROT 0 DO \ k Schleifendurchläufe
    DUP 1 AND \ Hole 1-Bit von n heraus
    ROT 2* + \ n' um 1 nach links und 1-Bit hinzufügen
    SWAP 2/ \ n um 1 nach rechts
  LOOP DROP \ Wenn Schleife beendet, n verwerfen
;

```

Wie funktioniert das? Das Unterprogramm erwartet auf dem Stack eine Ganzzahl **n** im Bereich

$$0 \leq n \leq 2^k = N$$

wobei **N** die Ordnung der FFT (Zweierpotenz) bezeichnet. Die Schleife muß  $k = \log_2[N]$  mal durchlaufen werden, die Schleifengrenzen liegen also bei 0 und **k**. Der Einfachheit halber wird **k** auf dem Stack unmittelbar vor **n** angeordnet, und nicht aus einer Variablen geholt. Um einzusehen, wie das Unterprogramm die Reihenfolge der Bits umkehrt, stelle man sich die (Eingabe-) Ganzzahl **n** in Binärschreibweise vor: Eine Zeichenkette von Einsen und Nullen in einem **k** Bit langen Feld. Wenn die Ordnung der FFT beispielsweise 16 ist, dann ist das Feld **k=4** lang. Die Zahl 7 wird beispielsweise als

$$n = 7_a = 0111_b$$

dargestellt, und dieselbe Zahl in umgekehrter Bitreihenfolge ist

$$n' = 1110_b = 14_a$$

Wir beginnen mit  $n' = 0$  (alle Bits auf 0 gesetzt). Dann verschieben wir  $n'$  eine Stelle nach links und addieren das am weitesten rechts stehende Bit von **n**. Schließlich verschieben wir **n** um eine Stelle nach rechts, wobei das ursprünglich am weitesten rechts stehende Bit von **n** herausfällt. Und das wiederholen wir, bis wir fertig sind.

Die Verschiebeoperationen simulieren wir über Ganzzahldivision durch 2 ( $/$ ) für die Verschiebungen nach rechts, und Multiplikation mit 2 ( $2*$ ) für die Verschiebungen nach links. **n** und  $n'$  bewahren wir auf dem Datenstack auf (was auf temporäre lokale Variablen hinausläuft, die wieder freigegeben werden, sobald das Unterprogramm die Regie an das Hauptprogramm zurückgibt). Wie wir es in Forth gewöhnt sind, überprüfen wir das sofort:

```

4 7 STIB . 14 ok
4 14 STIB . 7 ok

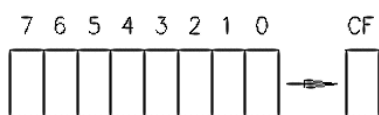
```

Ein Maschinenprogramm, das die Operationen vollständig innerhalb der CPU-Register ausführt, ist wesentlich schneller als das eben beschriebene High-Level-Programm. Dreh- und Angelpunkt sind dabei die Befehle **SHR** ("Logical Shift Right") und **RCL** ("Rotate Left Through Carry"). Sie führen zu einem extrem einfachen Programmstück, dessen Wirkungsweise wir uns verdeutlichen

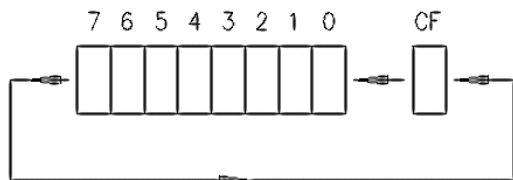


# Assemblieren leicht gemacht

wollen (siehe Bild).



SHR



RCL

Die genaue Vorgehensweise hängt von dem jeweiligen Forth-System ab. Mehrere kommerzielle Systeme bewahren den obersten Stackeintrag im Register **BX** auf, was einige **PUSH**- und **POP**-Operationen einspart. Das Public-Domain-System F-PC läßt **BX** frei. Wir gehen in unserer Veranschaulichung von F-PC aus und besorgen uns zunächst einmal den Eingabewert **n**. Wir **POP**pen ihn vom Stack und legen ihn nach **BX**:

```
POP BX
```

Als nächstes bestimmen wir das (frei zur Verfügung stehende) Register **DX** zur Aufnahme des Endergebnisses unserer Bitfolgeumkehrbemühung. Wir belegen **DX** mit dem Anfangswert **0**, was wir über das bitweise exklusive Oder ganz leicht erreichen können [11]:

```
XOR DX, DX
```

Jetzt verschieben wir (den Inhalt von) **BX** über **SHR** um eine Stelle nach rechts. Wie wir aus dem Bild ersehen, wird das Bit ganz rechts aus dem Register in das Carry-Flag geschoben. Dann verschieben wir **DX** mittels **RCL** um eine Stelle nach links. Das im Carry-Flag befindliche Bit wird nun das Bit ganz rechts in **DX**. Das Bit ganz links in **DX** (wenn da überhaupt etwas drin ist) landet im Carry-Flag (CF). Das hat aber nichts zu besagen, da es ja dann gleich wieder durch das neue Bit ganz rechts in **BX** ersetzt wird, sobald wir die Befehlsfolge wiederholen. Das Maschinenspracheprogramm (mit Kommentaren) sieht also wie folgt aus:

```

\ Initialisierungsschritte
POP BX      \ n holen
XOR DX, DX  \ n'=0
\ Die nächsten Befehle k-mal wiederholen
SHR BX, 1   \ Logisch Rechtsverschieben um 1 Stelle
RCL DX, 1   \ Linksrotieren über Carry-Bit um 1 Stelle
```

Jetzt müssen wir noch dafür sorgen, daß das Befehlspar in den letzten beiden Zeilen die vorgesehene Anzahl von Wiederholungen durchläuft. Der Einfachheit halber erledigen wir das mit der einfachsten aller Schleifenoperationen, mit **LOOP**. Dazu müssen wir das Register **CX** mit der Anzahl der angestrebten Schleifendurchläufe laden und am Ende der Schleifenzeilen den **LOOP**-Befehl ausführen. Der **LOOP**-Befehl setzt den Wert von **CX** um **1** herab und springt an den Anfangspunkt der Schleife (den wir noch irgendwie bezeichnen müssen - wir kommen gleich darauf zu sprechen) zurück, solange **CX** noch nicht **0** ist. Das heißt, die Schleife wird genau so oft durchlaufen, wie die Zahl in **CX** angibt.

Wollen wir dieses Programmstück mit Hilfe eines Assemblers wie MASM oder TASM assemblieren, so müssen wir eine Textdatei der folgenden Form vorbereiten:

```

POP BX      ; n holen
POP CX      ; Zahl der Schleifendurchläufe holen
XOR DX, DX  ; n' = 0 setzen
HERE:      ; Schleifenanfang
SHR BX, 1   ; Nulltes Bit von n nach CF schicken
```



```

                ; und Rechtsverschiebung um 1 Stelle
RCL DX, 1      ; n' nach links verschieben und
                ; nulltes Bit von n' mit CF füllen
LOOP HERE     ; CX=CX-1; an Schleifenanfang, falls CX noch nicht 0
PUSH DX       ; Ergebnis auf dem Stack ablegen
    
```

Wir müssen jedoch noch ein paar Rahmenzeilen hinzufügen, die sich nach dem jeweiligen Assembler richten, und ein paar Bedingungen beachten, um das Programmstück von der verwendeten Hochsprache aus als Unterprogramm aufrufen zu können.

Um das Assemblerprogramm mit dem in F-PC eingebauten Assembler zu testen, ändern wir es ein ganz klein wenig ab, um uns an die Schreibweise des F-PC-Assemblers anzupassen. Wie erhalten:

```

CODE STIB      \ Bitreihenfolge umkehren
POP BX         \ n holen
POP CX         \ Zahl der Schleifendurchgänge holen
XOR DX, DX     \ n' = 0 setzen
HERE          \ Schleifenanfang
SHR BX, # 1    \ Nulltes Bit von n nach CF schicken
                \ und Rechtsverschiebung um 1 Stelle
RCL DX, # 1    \ n' nach links verschieben und
                \ nulltes Bit von n' mit CF füllen
LOOP           \ CX=CX-1; an Schleifenanfang, falls CX noch nicht 0
PUSH DX        \ Ergebnis auf dem Stack ablegen
NEXT END-CODE \ CODE-Definition beenden
    
```

Ein in Forth geschriebener Assembler ist einfach. Die (durch geeignete Bezeichnung an ihre Funktion erinnernden) Assembler-Befehle sind **IMMEDIATE**-Worte, die schon während des Assemblierens ausgeführt werden. Sie legen den entsprechenden Opcode in das Parameterfeld der im Aufbau befindlichen **CODE**-Definition. In F-PC erwarten die **LOOP**-Opcodes (**LOOP**, **LOOPZ**, **LOOPNZ** usw.) eine Zahl auf dem Stack. Es handelt sich dabei um diejenige Adresse, zu welcher sie zurückspringen (oder auch nicht, was davon abhängt, ob die gestellte Bedingung erfüllt ist oder nicht). Diese Adresse (auf dem Stack) kann über ein explizit angegebenes Label erzeugt werden. Wir können dazu aber auch, wie im obigen Beispiel, einfach nur **HERE** sagen, wodurch die Anfangsadresse des darauffolgenden und noch zu assemblierenden Maschinencodestücks auf den Stack gelegt wird. Das ist dann genau die Stelle, zu der wir zurückspringen wollen. Mit anderen Worten, **LOOP** assembliert den Intel-Befehl **LOOP** mit der dazugehörigen Rücksprungadresse.

Wir geben jetzt die besagte **CODE**-Definition von der Tastatur aus ein und überprüfen das Ergebnis.

```

CODE STIB ok
POP BX ok
POB CX POB <-WHAT?
    
```

Hoppla! Ein Tippfehler! Das Ganze nochmal! Bereinigen mit **FORGET** von **STIB** an:

```

FORGET STIB ok
CODE STIB ok
POP BX ok
POP CX ok
XOR DX, DX ok
HERE ok
SHR BX, # 1 ok
RCL DX, # 1 ok
LOOP ok
PUSH DX ok
NEXT ok
END-CODE ok
    
```



## Assemblieren leicht gemacht

Das scheint jetzt in Ordnung zu sein - der Assembler hat sich jedenfalls nicht beschwert. Probieren geht jedoch über Studieren:

```
4 7 STIB . 14 ok
4 14 STIB . 7 ok
```

Hurra! Diesmal haben wir es geschafft.

Will man eine Assembler-Version von **STIB** mit einem BASIC- oder C-Programm verbinden, so hat man noch ein paar kleinere Anpassungen vorzunehmen:

- Die Kommentare müssen da, wo Forth traditionell den umgekehrten Schrägstrich "\ " verwendet, von einem Semikolon ";" eingeleitet werden.
- Anstelle des Forth-Wortes **HERE** muß ein Schleifen-Label stehen.
- Der in BASIC oder C übliche Kopfteil muß hinzugefügt werden und der Definitions-Abschluß ändert sich auch.

Als Endergebnis, das mit einem der üblichen eigenständigen Assembler verarbeitet werden kann, erhalten wir:

```
Code segment word public 'CODE' ; Codesegment festlegen, assume cs:
Code public STIB                ; Allen Routinen Zugriff erlauben
  STIB proc near                 ; Bitreihenfolge umkehren
  POP BX                         ; n holen
  POP CX                         ; Zahl der Iterationen holen
  XOR DX, DX                     ; n' auf 0 setzen
HERE:                            ; Label, Schleifenanfang
  SHR BX, 1                      ; Nulltes Bit CF, eins nach rechts
  RCL DX, 1                      ; n' nach links, CF nulltes Bit
  LOOP HERE                     ; CX=CX-1, Schleife bei CX nicht 0
  PUSH DX                       ; Ergebnis auf Stack legen
  RET                            ; Rückkehr vom Funktionsaufruf
  STIB endp                     ; Ende der Definition
Code ends
end
```

- [6] Zumindest alle im Handel befindlichen Forths und auch viele aus der Public Domain.
- [7] Solche Assembler-Anweisungen, wie Makros und Pseudo-Opcodes, sind natürlich keine wirklichen Maschinenbefehle.
- [8] Siehe z.B. M. Abrash, *The Zen of Code Optimization* (The Coriolis Group, Inc., Scottsdale, AZ, 1994), wo in beredter Weise die Programmierung in Assembler gegenüber der in "höheren" Sprachen verteidigt wird.
- [9] Wortspiel im Englischen: "to assemble" = "assemblieren", aber auch "sich versammeln". Ein Recht, das in der amerikanischen Verfassung verankert ist, in der englischen nicht (engl. Red.)
- [10] Das bedeutet **BITS** rückwärts buchstabiert. Forth-Namen machen oft einen seltsamen Eindruck auf Programmierer, die an die barocken Namenszusammensetzungen von Funktionen in C gewöhnt sind. Die Forth-Gepflogenheiten haben selbstdokumentierenden Code zum Ziel, mit Namen im Telegrammstil, die die Funktionsweise ohne lange Nebenbemerkungen ausdrücken. Man hätte auch **BIT\_REV** nehmen können, was vielleicht weniger geheimnisvoll gewesen wäre
- [11] Man könnte auch den Befehl **MOV DX, # 0** verwenden, bräuchte dazu dann aber 1 Byte mehr an Speicherplatz.

*Dr. Julian Noble ist Professor für Physik an der Universität von Virginia. Den Mitgliedern der FIG UK mit starkem Langzeitgedächtnis wird noch seine Konstruktion einer Einsteckkarte mit dem 32-Bit-Fließkomma-Coprozessor AMD 9511 für den Jupiter Ace aus dem Jahre 1983 in Erinnerung sein. Die Auftraggeber und Jupiter Cantabs brachen beide etwa zur selben Zeit zusammen. Die ganze Angelegenheit, vom Entwurf bis zur Auslieferung, nahm nur drei Wochen Zeit in Anspruch (engl. Red.) Für Mathematiker und Naturwissenschaftler interessant ist Julian Nobles Buch "Scientific Forth", das in den Forth-Gemeinden der Welt natürlich bestens bekannt ist (Übers.).*





# Elementare Sortieralgorithmen

Bernd Beuster <bernd.beuster@epost.de>

23. Februar 2002

## 1 Überblick

In diesem Artikel werden einfache Sortieralgorithmen beschrieben, welche einfach zu implementieren sind und für Vektoren kleiner bis mittlerer Größe die Performance von komplizierteren Algorithmen wie Quick Sort übertreffen können. Die Algorithmen wurden mit der Linuxversion von bigForth entwickelt, sollten aber mit jedem anderen ANS-Forth funktionieren.

## 2 Algorithmen und Eigenschaften

### 2.1 Selection Sort

Bei dem Selection Sort-Algorithmus wird im nichtsortierten Teil des Vektors nach dem jeweils kleinsten Element gesucht und dieses mit einem Schritt an seine Endposition verschoben (Abb. 1).

Die Laufzeit von Selection Sort beträgt  $O(N^2)$ . Da jedes Element nur einmal verschoben wird ( $O(N)$ ), hat Selection Sort Vorteile, wenn die Elementanzahl klein ist und die Größe der einzelnen Elemente sehr groß ist. Dann ist die Laufzeit dieses Algorithmus  $O(N)$ .

### 2.2 Insertion Sort

Bei Insertion Sort wird das links liegende Element des nichtsortierten Teils des Vektors in den sortierten Teil eingefügt (Abb. 2).

Nachteilig ist, daß nur benachbarte Elemente miteinander verglichen werden, so daß weit entfernte Elemente eine lange Zeit benötigen, um an ihre Endposition zu gelangen.

Die Laufzeit von Insertion Sort beträgt  $O(N)$ , wenn der Vektor bereits sortiert ist, sonst  $O(N^2)$ .

### 2.3 Bubble Sort

Bei Bubble Sort wird von rechts im nichtsortierten Teil des Vektors beginnend das jeweils kleinste Element nach links zum sortierten Vektorabschnitt verschoben (Abb. 3).

Die Laufzeit von Bubble Sort beträgt  $O(N^2)$ .

### 2.4 Shell Sort

Shell Sort wurde 1959 von Shell erfunden. Es ist ein Insertion Sort, bei dem die Schrittweite  $h$  veränderlich ist. Dadurch können weit entfernte Vektorelemente mit wenigen Schritten in die Nähe ihrer Endposition verschoben werden (Abb. 4).

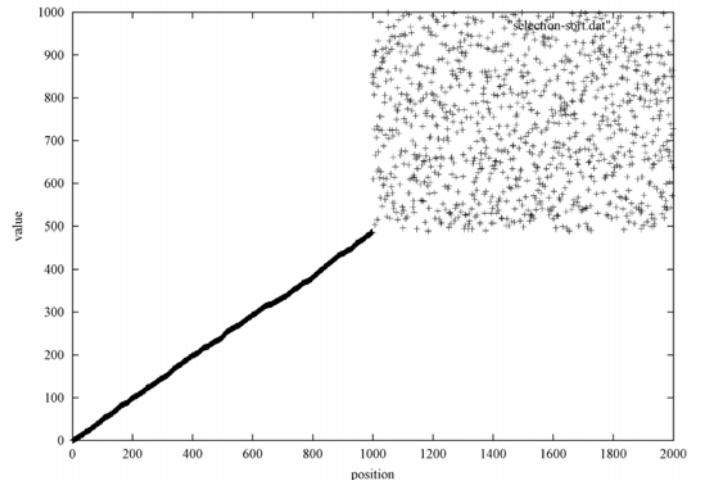


Abbildung 1: Selection

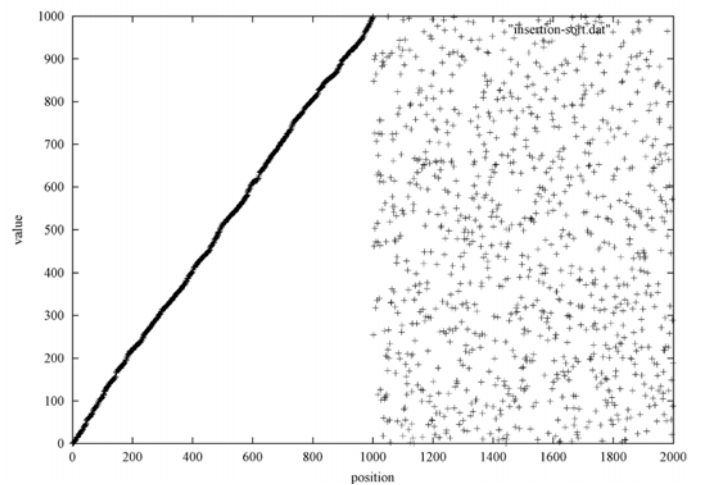


Abbildung 2: Insertion Sort

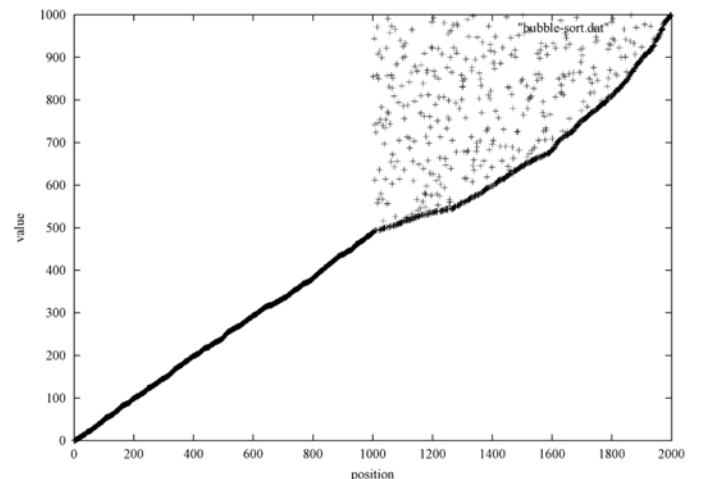


Abbildung 3: Bubble Sort

Die Schrittweitensequenz war bei dem Originalalgorithmus 1, 2, 4, 8, 16, . . ., diese ist aber nicht optimal (siehe [1]). Besser ist eine Sequenz von 1, 4, 13, 40, 121, 364, 1093, 3280, 9841, . . . mit einer Laufzeit von maximal  $O(N^{1.5})$  bzw.  $O(N^{1.25})$  bei zufällig verteilten Elementen.



## 3 Benchmarks

### 3.1 Methode

Zur Verifizierung wurde die Laufzeit der Algorithmen gemessen. Dabei wurde das TSC-Register der CPU (Pentium 4/1.5 GHz) verwendet.

Die Messung erfolgte mit einem zufällig sortierten und einem bereits sortierten Vektor. Es wurden jeweils drei Messungen durchgeführt, wobei der Median als Mittelwert verwendet wurde.

Mittels Gnuplot wurden die Meßwerte gefittet und der Multiplikator  $a$  der Laufzeitformel  $a \cdot N^b$  ( $b = \{1, 1.25, 2\}$ ) berechnet.

### 3.2 Ergebnisse

Die Meßergebnisse sind in Tab. 1 zusammengefaßt.

N	Selection		Insertion		Bubble		Shell	
	zufäll.	sort.	zufäll.	sort.	zufäll.	sort.	zufäll.	sort.
	$11.4 \cdot N^2$	$11.3 \cdot N^2$	$12.3 \cdot N^2$	$79.4 \cdot N^1$	$21.7 \cdot N^2$	$11.5 \cdot N^2$	$254 \cdot N^{1.25}$	$112 \cdot N^{1.25}$
$1 \cdot 10^1$	$3.0 \cdot 10^3$	$2.6 \cdot 10^3$	$2.6 \cdot 10^3$	$1.1 \cdot 10^3$	$3.0 \cdot 10^3$	$2.5 \cdot 10^3$	$6.9 \cdot 10^3$	$4.6 \cdot 10^3$
$2 \cdot 10^1$	$8.5 \cdot 10^3$	$7.2 \cdot 10^3$	$6.5 \cdot 10^3$	$1.9 \cdot 10^3$	$1.0 \cdot 10^4$	$6.4 \cdot 10^3$	$1.2 \cdot 10^4$	$7.6 \cdot 10^3$
$5 \cdot 10^1$	$3.9 \cdot 10^4$	$3.6 \cdot 10^4$	$3.7 \cdot 10^4$	$4.3 \cdot 10^3$	$5.4 \cdot 10^4$	$3.2 \cdot 10^4$	$4.1 \cdot 10^4$	$2.3 \cdot 10^4$
$1 \cdot 10^2$	$1.3 \cdot 10^5$	$1.3 \cdot 10^5$	$1.3 \cdot 10^5$	$8.4 \cdot 10^3$	$2.0 \cdot 10^5$	$1.2 \cdot 10^5$	$9.2 \cdot 10^4$	$4.5 \cdot 10^4$
$2 \cdot 10^2$	$5.0 \cdot 10^5$	$4.9 \cdot 10^5$	$5.0 \cdot 10^5$	$1.7 \cdot 10^4$	$7.8 \cdot 10^5$	$4.9 \cdot 10^5$	$2.1 \cdot 10^5$	$1.1 \cdot 10^5$
$5 \cdot 10^2$	$2.9 \cdot 10^6$	$3.0 \cdot 10^6$	$3.2 \cdot 10^6$	$4.1 \cdot 10^4$	$5.0 \cdot 10^6$	$2.8 \cdot 10^6$	$7.1 \cdot 10^5$	$3.4 \cdot 10^5$
$1 \cdot 10^3$	$1.1 \cdot 10^7$	$1.1 \cdot 10^7$	$1.2 \cdot 10^7$	$8.0 \cdot 10^4$	$2.0 \cdot 10^7$	$1.1 \cdot 10^7$	$1.6 \cdot 10^6$	$7.1 \cdot 10^5$
$2 \cdot 10^3$	$4.6 \cdot 10^7$	$4.5 \cdot 10^7$	$4.8 \cdot 10^7$	$1.6 \cdot 10^5$	$8.4 \cdot 10^7$	$4.4 \cdot 10^7$	$3.8 \cdot 10^6$	$1.7 \cdot 10^6$
$5 \cdot 10^3$	$2.8 \cdot 10^8$	$2.8 \cdot 10^8$	$3.0 \cdot 10^8$	$4.0 \cdot 10^5$	$5.4 \cdot 10^8$	$2.8 \cdot 10^8$	$1.1 \cdot 10^7$	$4.9 \cdot 10^6$
$1 \cdot 10^4$	$1.1 \cdot 10^9$	$1.1 \cdot 10^9$	$1.2 \cdot 10^9$	$7.9 \cdot 10^5$	$2.2 \cdot 10^9$	$1.2 \cdot 10^9$	$2.5 \cdot 10^7$	$1.1 \cdot 10^7$

### sortings.fs

```
marker empty
\s" selection-sort.fs" included
\s" insertion-sort.fs" included
\s" bubble-sort.fs" included
\s" shell-sort.fs" included
s" shell-sort-asm.fs" included

10000 constant N
N cells allocate throw constant array

: !array array N 1- for 1000 random over ! cell+ next drop ;
: .array array N 1- for dup @ cr . cell+ next drop ;

\ serialize instructions
code 0ticks ( -- ) ax push ax xor bx push cpuid bx pop
ax pop next end-code

\ read time-stamp counter
code ticks ( -- ud ) ax push rdtsc ax push dx ax mov next
end-code
```

Bubble Sort und Selection Sort haben die schlechtesten Laufzeiteigenschaften. Bei wenigen Elementen ( $20 \dots 50$ ) hat Insertion Sort Vorteile gegenüber Shell Sort.

## 4 Schlußfolgerungen

Wenn nur wenige Elemente zu sortieren sind, insbesondere wenn die Vektorelemente schon vorsortiert sind, ist *Insertion Sort* zu bevorzugen. Bei moderat großen Vektoren sollte *Shell Sort* bevorzugt werden.

### Literatur

[1] Robert Sedgewick, Algorithms in C (Parts 1-4), Third Edition, Addison-Wesley Publishing Company Inc., ISBN 0-201-31452-5, 1998.

: benchmark 0ticks ticks array N sort ticks 2swap d- cr ud. ;

```
\ Benchmark the sorting algorithms
!array benchmark
!array benchmark
!array benchmark
cr
benchmark
benchmark
benchmark
cr
Bye
```

### bubble-sort.fs

```
\ For each i from 0 to N-1, th inner loop puts the minimum
\ element among the elements in a[i],...,a[N-1] into a[i] by
\ passing from right to left through the elements,
\ compare-exchanging successive elements.
\ The smallest one moves on all such comparisons, so it
\ "bubbles" to the beginning. As in selection sort, as the
\ index i travels from left to right through the file, the
```



\ elements to its left are in their final position in the array.

```
\ compare values in a,a-1 and exchange
: ?exchange ( a -- a ) dup cell- over @ over @
  2dup < if >r swap ! r> over ! exit then 2drop drop ;
```

```
\ inner loop
: (sort) ( a u -- ) for ?exchange cell- next drop ;
```

```
: sort ( a u -- ) 1- 2dup cells + swap 1- dup
  for 2dup (sort) rot cell+ -rot 1- next 2drop drop ;
```

## insertion-sort.fs

\ For each i from 0 to N-1 sort the elements a[0],...,a[i] by  
 \ moving one position to the right elements in the sorted list  
 \ a[0],...,a[i-1] that are larger than a[i], then putting a[i] into  
 \ its proper position.

```
\ compare values in a,a-1 and exchange
\ return 0 if finished
: ?exchange ( a -- a ) dup cell- over @ over @
  2dup < if >r swap ! r> over ! exit then 2drop drop 0= ;
```

```
\ inner loop
: (sort) ( a u -- )
  for ?exchange dup while cell- next else unloop then drop ;
```

```
: sort ( a u -- ) swap cell+ 0 rot 2-
  for 2dup (sort) swap cell+ swap 1+ next 2drop ;
```

## selection-sort.fs

\ For each i from 0 to N-1 exchange a[i] with minimum  
 \ element in a[i],...,a[N-1]. As the index i travels from left to  
 \ right, the elements to its left are in their final position in the  
 \ array (and will not be touched again), so the array is fully  
 \ sorted when i reaches the right end.

```
\ compare values in a1,a2 and return address of the smaller
\ one
: less ( a1 a2 -- a3 ) over @ over @ > if swap then drop ;
```

```
\ inner loop
: (sort) ( a1 u -- a2 ) over swap for cell+ tuck less swap next
  drop ;
```

```
\ exchange values in addresses a1,a2
: exchange ( a1 a2 -- ) over @ over @ >r swap ! r> swap ! ;
```

```
: sort ( a u -- ) over swap 2- dup
  for dup >r over >r (sort) r@ exchange r> cell+ r> 1- next
  2drop drop ;
```

## shell-sort.fs

\ If we change Insertion Sorts step width from 1 to h we get a  
 \ h-sorted file. Adding an outer loop to change the  
 \ increments leads to this implementation, which uses the  
 \ increment sequence  
 \ 1,4,13,40,121,364,1093,3280,98411,...

```
variable h
: !h ( n -- ) 9 / 1 begin 2dup >= while 3 * 1+ repeat h !
  drop ;
: h+ ( a1 -- a2 ) h @ cells + ;
: h- ( a1 -- a2 ) h @ cells - ;
: h/ ( u1 -- u2 ) h @ / ;
```

```
\ compare values in a,a-h and exchange
\ return 0 if finished
: ?exchange ( a -- a ) dup h- over @ over @
  2dup < if >r swap ! r> over ! exit then 2drop drop 0= ;
```

```
\ inner loop
: (sort) ( a u -- )
  for ?exchange dup while h- next else unloop then drop ;
```

```
: ins-sort ( a u -- ) swap h+ 0 rot h @ - 1-
  for 2dup h/ (sort) swap cell+ swap 1+ next 2drop ;
```

```
: sort ( a u -- ) dup !h begin 2dup ins-sort h @ 3 / dup h ! 0=
  until 2drop ;
```

## shell-sort-asm.fs

\ assembler version of inner loop

```
variable h
: !h ( n -- ) 9 / 1 begin 2dup >= while 3 * 1+ repeat h ! drop ;

: h+ ( a1 -- a2 ) h @ cells + ;
```

```
code h/ ( u1 -- u2 ) h #) cx mov dx dx xor cx idiv next end-
code
```

```
\ inner loop
code (sort) ( a u -- )
  cx pop si push di push bx push
  cx si mov h #) cx mov \ ax=u si=a cx=h/4
  si di mov cx cx add cx cx add \ di=a cx=h
  si ) bx mov \ bx=[a]
  begin
  cx di sub \ a-h
  di ) bx cmp \ [a-h]<[a] ?
  < while
  di ) dx mov \ move array up
  dx 4 di D) mov
  4 # si add \ a=a+4
  1 # ax sub \ u=u-1
  U< until
  then
```



## aus dem Netz

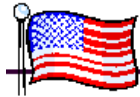
```
bx di cx I) mov
bx pop di pop si pop ax pop next end-code
```

```
: ins-sort ( a u -- ) swap h+ 0 rot h @ - 1-
  for 2dup h/ (sort) swap cell+ swap 1+ next 2drop ;
```

```
: sort ( a u -- ) dup !h begin 2dup ins-sort h @ 3 / dup h ! 0=
until 2drop ;
```

Bernd Beuster

Mehr von Henry...



Grüße, an Euch alle!

Es ist Samstagabend in Garmisch und ich wünschte, ich könnte mich zur Deutschen Forth Gruppe mit einem vollen Maß Bayerischen Biers gesellen.

Stattdessen muß ein herzliches 'Prosit' aus Kalifornien genügen! Ich bin sicher, daß ich Euch allen anläßlich der jährlichen deutschen Forth-Konferenz die Grüße und besten Wünsche der Silicon Valley Forth Gruppe übermitteln darf.

Unser April-Treffen fand zeitig statt, letzten Samstag. Da uns meine Tochter aus Deutschland besuchte, konnte ich nicht teilnehmen. Ich schaute jedoch für ein paar Minuten am Morgen herein, um die letzte Ausgabe (#116) der Forthwrite abzugeben, die zwei Tage vorher eintraf.

Dr. Ting war schon da, bereit eine Beschreibung seines F# (F-sharp, in englisch) zu geben. Dies ist eine 32-Bit, Protected Mode, Subroutine-Threaded eForth Implementation mit einem Windows Interface.

Gemäß dem Tagungsprogramm sollte Jeff Fox eine kurze Präsentation seines aktuellsten GUI Desktop vorführen. Da er, wie ich hörte, einer der häufigeren SVFIG-Besucher auf der FIG UK Webseite ist, stehen die Chancen gut, daß er dort einige Informationen hinterlassen wird.

Rollt das Faß 'raus und habt eine Menge Spass!

Grüße an alle,

Henry

*Henry, den Spaß hatten wir, mit und ohne Bier, hoch in den Bergen, nahe der Zugspitze und in einem Hotel, das nur über eine Seilbahn zu erreichen ist. Die Seilbahn arbeitet ab dem frühen Abend nicht mehr. Der Drachen hatte seinen Spaß an uns und wir alle miteinander.*

fep

[... zurück zu Seite 8 ...](#)

...gute Nachricht ...



Betreff: [ANN] pbForth Version 2.1.0

Release with USB Support

Datum: Wed, 10 Apr 2002 10:42:23 -0400

Von: "pbForth" <pbforth@hempeldesigngroup.com>

Firma: Hempel Design Group

An: "Friederich Prinz" <Friederich.Prinz@t-online.de>

For everyone that has been interested in firmware speed over the last few days, I am **ready to announce the latest version of pbForth.**

**This version supports the USB tower under Windows !!!!**

An empty loop of 20,000 iterations takes less than 1 second with pbForth, and you get a fully interactive development environment, with inline assembler support coming shortly, thanks to Lugnet contributors.

Major changes include:

1. Support for USB tower under Windows
2. Serial port settings changed to 2400,0,8,1 (odd parity)
3. Passes full ANSI test suite of 800 tests
4. Kernel size is about 10K
5. Much more reliable communication

Please give it a try and note any installation problems. Report any problems to the robotics.rcx.pbforth group at

[<http://www.lugnet.com>](http://www.lugnet.com)

or send me an email.

I'll be updating docs and adding application notes, on the main pbForth weblog page at:

[<http://www.hempeldesigngroup.com/lego/pbForth>](http://www.hempeldesigngroup.com/lego/pbForth)

Cheers,

Ralph

### Wie funktioniert eigentlich GPS ?

Welche Signale kommen von den Satteliten? Wie „kriege ich die in den PC“ (die Signale, natürlich)? Was bedeuten die Signale? Wie kann ich aus den Signalen meinen Standort dreidimensional errechnen? Wie kann ich mir einen GPS-Empfänger selbst bauen und den dann auch noch mit Forth (womit sonst) programmieren?

Findet sich unter den Forthern Jemand, der das Thema bereits aufgearbeitet hat ? Platz für entsprechende, ausführliche Artikel bietet die VD genug.



Friederich Prinz



### Bericht aus Garmisch-Partenkirchen

Das war schon ein Glanzlicht der ganz besonderen Art, die Einladung zur Tagung nach Garmisch-Partenkirchen. Die Stadt ist an sich schon ebenso malerisch wie die Umgebung, in der sie liegt. Aber das war Ulrike und Heinz Schnitter gar nicht gut genug für die Mitglieder der Forthgesellschaft. Hoch hinaus auf eine Alm mußten wir; wo kein Auto hin kann (fast keines), wo niemand zu Fuß hinaufläuft (außer aus Versehen) und wo eine himmlisch forthige Ruhe uns gute Gespräche und forthiges Tun geradezu aufzwang.

Das Forsthaus Graseck (Forsthaus; nicht Forthhaus !) ist natürlich keine klassische Almhütte, sondern ein sehr großes und komfortables Haus, das Wellness für die Damen (Massagen, Kräutertees, Heupackungen) anbietet. Wellness für den Herren hat es natürlich auch auf dem Programm (Weißbier, Hirschgulasch, Zigarren). Und erreichbar ist das Haus über eine Seilbahn, welche die weltberühmte Partnachklamm überquert. Um den Parkplatz vor der Seilbahn zu erreichen, muß man sich als Automobilist allerdings über eine Reihe von Schildern hinwegsetzen, welche die Einfahrt in das Tal der Partnach nur dem Ortsansässigen erlauben. Aber im *Darüber hinweg setzen* sind Forther ja durchaus geübt.

Atemberaubend war, wie von Heinz versprochen, gleich bei der Ankunft das Panorma, bzw. der Blick auf die Wettersteinwand. Am Donnerstag hat es nämlich noch nicht geregnet ! Die leuchtend weißen Berge haben den ganzen, sonnigen Nachmittag über das Tal mit einem eigenen Licht erfüllt. Und in der Nacht haben Schnee und Eis im Sternenlicht so diffus geleuchtet, wie sonst nur der Konsens des Swap im Drachenrat über den Köpfen der Ratsmitglieder.

Das Angebot, einen Tag früher anzureisen und in Ruhe zu akklimatisieren, nehmen in jedem Jahr mehr Vereinsmitglieder wahr. Darum waren wir nicht nur am Donnerstagabend eine große und vergnügte Runde, sondern auch am Freitagmorgen eine recht große Gruppe, die sich mit einem Autokorso auf den Weg zum Walchensee gemacht hat. Einige der ganz harten Mädels und Burschen waren schon gleich am Donnerstagnachmittag zum ersten Mal auf einer Wanderung in den Bergen. Das war mir aber entschieden zu anstrengend. Die Fahrt auf den Herzogstand, per Seilbahn, versteht sich, gefiel mir da schon besser. Von der Bergstation, die noch weit vom Gipfel des Herzogstand entfernt ist, hatten wir einen wunderschönen Ausblick auf den Walchensee. Meine Frau konnte einen kleinen Schneemann bauen und mit Ute Woitzel eine Schneeballschlacht anzetteln (1 Schneeball, aus 2 m Entfernung, in Absprache zwischen Werferin und „Opfer“). Eine kleinere, beinharte Gruppe unserer Reisegesellschaft durfte sich für eine halbe Stunde auf den Spuren Messners im Himalaja fühlen und Schnee und Eis in Richtung des wolkenverhangenen Herzogstand queren. Die waren aber klug genug, beizeiten umzukehren.

Weniger klug, aus meiner Sicht, waren die drei Burschen, die auf die Seilbahn verzichtet haben und den Berg hinauf gelaufen sind. Aber die sind noch jung. Da haben die ein Recht auf Unfug ;-)

Nach dem Herzogstand hatte Heinz einen Besuch im Kraftwerk Walchensee auf dem Plan. Das Kraftwerk steht tatsächlich am „Strand“ des Kochelsees, bekommt aber das Wasser für seine acht Turbinen aus dem Walchensee, der rund 200 m höher liegt. Die Führung durch das Kraftwerk war schon eine spannende und anregende Sache.

Noch spannender war für mich die Fahrt über die berühmte (unter Motorradfahrern berüchtigte) Kesselbergstraße zurück nach Garmisch. Weil gerade der Regen eingesetzt hatte und die Temperatur dem Gefrierpunkt nahe kam, entstand in mir nicht der Wunsch, jetzt lieber auf dem Motorrad zu sitzen. Aber fahren muß ich dort auf jeden Fall noch auf meiner Maschine ;-)

Nachdem alle Teilnehmer des Ausflugs zurück im Forsthaus waren, und nachdem mittlerweile auch der größte Anteil der Freitags anreisenden Tagungsteilnehmer eingetroffen war, konnte die eigentliche Tagung mit ihrem Vortragsteil beginnen. Darüber wird aber Bernd Paysan gesondert berichten.

Am Freitagabend wurde es dann wieder gemütlich. Bei gutem Essen, stimulierenden Getränken und in angenehmer Gesellschaft konnten die Vereinskameraden mit einander plaudern, ernsthaft debattieren, oder einfach nur über die mißglückte Simulation lästern, die Heinz zu verantworten hatte.

Mittlerweile hatte sich nämlich herumgesprochen, daß die Berge, die wir tags zuvor noch sehen konnten, die jetzt aber von dicken Regenwolken verdeckt waren, gar nicht echt sind !

Heinz hatte zugegeben, als optischen Hintergrund der Balkonsicht des Hauses Graseck eine Hintergrundtask auf einem seiner ONF-Rechner im Beschleuniger in Garching definiert zu haben. Um von Scans oder Photos unabhängig zu bleiben, hatte Heinz aus dem Zerfall des Neutrinos eine Vollsimulation des Urknalls abgeleitet und die ganze Schöpfungsgeschichte neu rechnen lassen. Solange Heinz noch persönlich eingreifen und manipulieren konnte, lief die Simulation so wie von ihm geplant. Allerdings gerieten sowohl der ONF-Rechner als auch die darauf laufende Simulation außer Kontrolle, als Heinz sich mit uns auf der Alm aufhielt. Bereits ab dem Freitagmorgen zogen ungeplante Wolken auf, die ab dem Mittag zu dem bereits angesprochenen Regen führten.

Abends wollen einige Tagungsteilnehmer sogar große Fische gesehen haben, die sich langsam von rechts nach links, also in östlicher Richtung, vor der Gebirgskulisse hinweg bewegt haben. Die Diskussion konnte keine Klarheit darüber erbringen, ob es sich bei den Fischen um einen weiteren Fehler in der Simulation handelte, oder einfach nur um einen der Bildschirmschoner in Garching.

Immerhin waren am Samstagvormittag wieder vereinzelte und zeitlich sehr begrenzte Blicke auf die Berge möglich, was den Tagungsteilnehmern aber kaum auffallen konnte. Das Programm war bis zum Mittag voll mit anspruchsvollen und anregenden Vorträgen, über die wieder Bernd berichten wird.

Den Nachmittag haben dann die meisten Tagungsteilnehmer wieder „außerhalb“ verbracht. Schon fast selbstverständlich durch Regen hindurch und über die erwähnte, faszinierende Kesselbergstraße ging es nach Kochel. Dort haben wir das Franz Marc Museum besucht.

... weiter auf Seite 31 ...



## Laudatio auf den Drachenbewahrer

Edle Herren, edle Frouwen,  
tapfere Recken, holde Maiden,  
erlauchte Gäste aus fernen Landen!

Gestern Abend zu später Stund'  
saßen zusammen in magischer Rund'  
alljene aus unserem erlauchten Kreis'  
die gehütet den Drachen – als Ehrenpreis.

Sie saßen zusammen fast bis in den Tag  
darüber ich berichten mag:

Was all dort beschlossen, will ich itzt und all hier euch berichten und kundtun.

Der EINE war zugegen – mehr zu sagen verbietet mir der Runde Gelübde – doch soviel sei verraten wohl: er lenkte gnädig und gerecht den Verstand der edlen Herren und der holden Frouwe.

Zwar versuchte die zierliche Maid Ute geheim zu erlauschen der Edlen Rede, doch wurde entdeckt ihr frevlisches Tun und sie verwiesen des Saales.

Mit magischen Künsten und uraltem, weisem Trunk klärte der gerechte SWAP der Anwesenden Sinn:

So dachten wir lange vor uns hin:  
wer sollt das heilige Ding' erwerben,  
wer sollt dies Jahr den Drachen erben?

Ich steh' itzt hier und verkünde die Kür:

Gar viele Mannen kundig in mancherlei Ding' standen an den Preis zu erhalten. Doch tat sich einer besonders hervor. Schon als Knappe bezwang er gar manch höllisch' Bit und Byte; er zähmte mit festem Griff den wilden Interruptus.

Nach abenteuerlicher Wanderschaft trat er in die Dienste eines Edlen des Forth und erfocht sich dort gar glänzenden Ruhm. Viele edle Klein(di)odien aus Okzident und Orient fügte er mit Elbengeschick und Zwergenkunst mit sicherer Hand zu kostbarem Geschmeide, das die Welt des Forthes nun zieret.

So erhebe dich nun tapferer Recke

**H**ANS **G**UCKES

Dir ist verliehen der Swap von edlem Blut –  
bewahre ihn sicher, behüte ihn gut.



Die Besucher hatten offensichtlich (fast) allesamt große Freude an diesem kulturellen Angebot. Ich persönlich hätte besser bei Egmont und Thomas (und vermutlich bei einem Bier) in Haus Graseck bleiben sollen.

Der Samstagabend brachte dann die übliche Kurzweil wie Abendessen, Drachenrat, Workshops (*hat da jemand teilgenommen? Berichte bitte!*) und angenehme Unterhaltungen über Heinz's Simulation und Anderes.

Der Sonntag brachte natürlich die Versammlung der anwesenden Mitglieder der FG mit sich. Über die Versammlung wird der Protokollführer berichten, **Ewald Rieger**.

Bemerkenswert finde ich aber, daß sich gleich zwei Freunde im Forth angeboten haben, die nächste Tagung auszurichten. Ich erinnere mich gut an Versammlungen, in denen es der Mitgliedschaft sehr schwer fiel, auch nur einen Organisator für die jeweils nächste Tagung zu finden. Heute haben wir die kommenden beiden Jahre bereits „im Sack“. Ich freue mich bereits darauf, im kommenden Jahr in die Gegend von Mannheim zu fahren, wohin Ewald Rieger uns einladen wird. Und im darauf folgenden Jahr wird **Uli Hoffmann** uns auf die Insel Fehmarn rufen, wo das Wetter im April mindestens so schlecht sein soll, wie in Garmisch-Partenkirchen ;-)

Zusammenfassend muß ich sagen, daß ich trotz der 700 km An- und Abfahrt jeden Tag genossen habe, auch wenn die Abende heute aus Altersgründen längst nicht mehr so lang sind, wie sei es früher einmal waren. Es war schön, die Freunde im Forth wieder zu sehen. Es war schön, daß so viele ihre Frauen mitgebracht haben. Es war gut, daß wir so viel miteinander zu lachen hatten.

**Ulrike und Heinz – ich danke Euch !**

*Friederich Prinz*



Photo: U.Hoffmann

*Ulrike Schnitter ; Thomas Beierlein vor Haus Graseck*

... zurück zu Seite 29 ...

## Assemblieren als Hobby ?

**Bernd Beuster**

*Der nachfolgende Beitrag von Bernd Beuster ist wegen seiner Korrespondenz zu dem Beitrag von Julian Noble für diese Ausgabe der VD ausgewählt worden. Der Artikel stammt ursprünglich aus dem Netz und wird mit der freundlichen Genehmigung des Autors hier wiedergegeben.*

Nach langem Zaudern habe ich mir einen neuen Rechner zugelegt (P5-MMX 233MHz/64MB EDO-RAM --> Pentium4 1.5GHz/256 RDRAM).

Die Entscheidung "Athlon oder P4" fiel zugunsten des P4, auch wegen seiner SSE2 Einheit. SSE2 ist eigentlich MMX auf 128 Bit aufgeböhrt und SSE mit double-floating point.

Das Gute an dem aufgeböhrtten MMX ist außerdem, daß die normalen Floatingpointregister nicht mehr mißbraucht werden, so daß in dieser Hinsicht keine Verzögerungen stattfinden.

Meine Experimente laufen unter bigForth, den Assemblercode für SSE2 erzeuge ich mit dem GNU Assembler -- soweit er dazu in der Lage ist, ansonsten schaue ich in der Intel-Docu nach.

### 1 Aufgabe

Gegeben sei ein RIFF/WAVE-File (z.B. die Windows Systemklänge unter WIN98), stereo, 16-Bit. Der Sound soll in Mono gewandelt werden.

### 2 Algorithmus

$$M := (L+R+1)/2$$

### 3 Implementierung

#### 3.1 Forth Hi-Level

```
code h@ ( a -- 16b ) ax ) ax movsx next end-code macro
: 2h@ ( a -- 16b1 16b2 ) dup h@ swap 2 + h@ ; macro
: v+ ( n1 n2 n3 n4 -- n5 n6 )>r rot + swap r> + ; macro

: >mono ( a1 u1 a2 u2 -- ) rot drop 4/ 1- for
over 2h@ + 1+ 2/ over 2dup w! 2 + w!
4 4 v+ next 2drop ;
```



## 3.2 Einfacher Assembler

```
code >mono ( a1 u1 a2 u2 -- )
ax cx mov dx pop ax pop ax pop \ eax=a1 edx=a2 ecx=u2
bx push di push
2 # cx shr
begin
ax ) bx movsx 2 ax d) di movsx \ ebx=L edi=R
di bx add 1 # bx add 1 # bx sar \ ebx=(L+R)/2 (rounded)
.w bx dx ) mov .w bx 2 dx d) mov
4 # ax add 4 # dx add
loop
di pop bx pop ax pop next end-code
```

## 3.2 Assembler und SSE2

Der eigentliche Befehl zur Mittelwertbildung 'pavgw' funktioniert nur mit unsigned Integer. Deswegen die Kopfstände mit Offset+/-.

Der Geschwindigkeitsvorteil ergibt sich wohl auch aus den 128-Bit Speicherzugriffen.

```
\ SSE2 needs 16 byte aligned data
: aligned ( a1 -- a2 ) 15 + -16 and ;
: align here dup aligned swap - 0 ?do 0 c, loop ;
```

```
create offs align $80008000 , $80008000 , $80008000 ,
$80008000 ,
```

```
code >mono ( a1 u1 a2 u2 -- )
ax cx mov dx pop ax pop ax pop \ eax=a1 edx=a2 ecx=u2
5 # cx shr
begin
\ load
$006f0ff3 , \ movdqu (%eax),%xmm0
$486f0ff3 , $10 c, \ movdqu 16(%eax),%xmm1

\ swizzle
$c0700ff3 , $d8 c, \ pshufhw $0xd8,%xmm0,%xmm0
$c9700ff3 , $d8 c, \ pshufhw $0xd8,%xmm1,%xmm1
$c0700ff2 , $d8 c, \ pshufw $0xd8,%xmm0,%xmm0
$c9700ff2 , $d8 c, \ pshufw $0xd8,%xmm1,%xmm1
$c0700f66 , $d8 c, \ pshufd $0xd8,%xmm0,%xmm0
$c9700f66 , $d8 c, \ pshufd $0xd8,%xmm1,%xmm1
$d06f0f66 , \ movq %xmm0,%xmm2
$d16d0f66 , \ punpckhqdq %xmm1,%xmm2
$c16c0f66 , \ punpcklqdq %xmm1,%xmm0

\ +offset
$05fd0f66 , offs aligned , \ paddw offs,%xmm0
$15fd0f66 , offs aligned , \ paddw offs,%xmm2

\ average
$c2e30f66 , \ pavgw %xmm2,%xmm0

\ -offset
$05f90f66 , offs aligned , \ psubw offs,%xmm0
```

```
$c86f0f66 , \ movq %xmm0,%xmm1

\ deswizzle
$c0610f66 , \ punpckhwd %xmm0,%xmm0
$c9690f66 , \ punpcklwd %xmm1,%xmm1

\ store
$02e70f66 , \ movntdq %xmm0,(%edx)
$4ae70f66 , $10 c, \ movntdq %xmm1,16(%edx)

32 # ax add 32 # dx add
-1 # cx add
nz while
repeat
ax pop next end-code
```

## 4 Ergebnisse

Mittels des Pentium-Befehls 'RDTSC' kann man die CPU-Takte bestimmen.

Der Algorithmus wurde dreimal durchlaufen, jeweils der minimale und der maximale Wert werden aufgeführt. Der langsame Wert spiegelt also auch den RAM-Zugriff dar, wogegen der schnellere Wert die Daten aus dem Cache widerspiegelt.

Clock cycles per sample (Pentium4, 1.5 GHz, RDRAM)

```
Hi-level : 50.4 - 59.4
Simple assembler : 8.9 - 16.1
Assembler and SSE2 : 7.2 - 14.8
```

## 5. Bemerkungen

Der Code ist sicher noch nicht 100%-ig optimal, aber ich bin schon froh, daß die SSE2-Version zumindest keine schlechteren Ergebnisse liefert.

Da der SSE2-Assembler-Code länger als 126-Byte ist, funktioniert ein

```
begin ... loop
```

bzw.

```
begin ... until
```

mit den strukturierten Assemblerbefehlen nicht mehr. Deswegen der Konstrukt

```
begin ... while repeat
```

da das REPEAT einen 'jmp'-Befehl erzeugt und dieser kann beliebig weit springen. (Ebenso könnte man "if...[else...]" durch "if else...[else...]" substituieren).

*Bernd Beuster*





## Forth-Tagung 2002 in Garmisch-Partenkirchen

Bernd Paysan

28. April 2002

Dieses Jahr fand die Forth-Tagung wieder ganz im Süden von Deutschland statt, in Garmisch-Partenkirchen, zum vierten Mal von Heinz und Ulrike Schnitter organisiert. Das Tagungshotel, über der Partnachklamm gelegen, war über die derzeit älteste Seilbahn Deutschlands erreichbar --- frisch gewartet und mit Forth-Programmiern getestet (als Testlast). Auch die malerische Aussicht vom Tagungshotel auf die Zugspitze war laut Heinz Schnitter "in Forth programmiert".

Am Donnerstag Nachmittag bestiegen die früh angereisten Forther den Eckbauer, ein 300 Meter höherer Hügel neben dem Tagungshotel. Das größere touristische Programm war für den Freitag Vormittag geplant. Zunächst ging es auf den Herzogstand (das G(eh-)Forth-Team zu Fuß, der Rest mit der Seilbahn), ein Berg, für dessen Berühmtheit König Ludwig persönlich gesorgt hat.

Danach besichtigten wir das Walchenseekraftwerk, das leistungsfähigste Wasserkraftwerk in Deutschland. 1924 in Betrieb genommen, besteht es --- bis auf ein paar Laptops im Leitstand --- noch aus originaler Technik, an der nur kleinere Wartungsarbeiten vorgenommen wurden.

Nach diesem Ausblick in die Stabilität elektrohydraulischer Technik, die auch von Forth-Programmen nicht erreicht wird, ging's mit den Vorträgen los.

**Anton Ertl** stellte in "Threaded Code Variations and Optimizations (Extended Version)" den aktuellen Stand der Technik für gefädelt Forth-Systeme vor: Primitiv-zentrischer Code, und eine Mixtur aus direkt und indirekt gefädelt Code. Colon-Definitionen sind direkt gefädelt, und rufen ausschließlich Primitives auf; falls doch mal ein High-Level-Wort ausgeführt wird, wird es mit dem Primitive call aufgerufen. Aufrufe mit execute dagegen werden indirekt erledigt --- genauer gesagt sogar doppelt indirekt, damit auch der Code von mit Create .. DOES> definierten Wörtern gefunden werden kann.

Mehrere Primitives können zu Superinstructions verbunden werden, das ist eine Art Peephole-Optimierung auf Threaded-Code-Ebene. Diese Superinstructions können sogar zur Laufzeit durch Zusammenkopieren der Primitives gebaut werden; damit hat man fast schon einen Native-Code-Compiler, nur die Sprünge werden noch nicht direkt umgesetzt.

**Hans Eckes** zeigte einen Untersatz zur Forth-Briefmarke, auf der ein USB-Chip war. Der bildet eine serielle Schnittstelle nach und ist damit sowohl auf PC-Seite als auch auf Target-Seite total trivial zu benutzen. Auf dem PC kann

man jedes normale Terminal-Programm verwenden, und sogar vergessen, welche Baudrate man verwendet, und was man mit den Parity-Optionen macht. Auf dem Target ist alles sogar noch einfacher, der Chip verhält sich einfach wie eine FIFO --- 'reinschreiben bis voll und 'rauslesen bis leer.

**Ulrich Hoffmann** produziert seine Dokumentation von Forth-Programmen jetzt mit einem Perl-Tool, POD ("Plain Old Documentation"). Das ist eine einfache Markup-Sprache, mit der man *HTML*, *nroff* und andere Formate generieren kann, und die man einfach zwischen den Code streut. Damit Forth damit zurecht kommt, braucht man nur die paar Start-Token von POD so zu definieren, daß sie den nächsten Start für Forth-Code suchen. Diese bewährte Technik hat mit POD also noch ein "Opfer" gefunden.

Am Samstag Morgen waren zunächst die beiden geladenen Gäste aus Holland (von der Forth-Gebriekersgroep) dran.

**Albert Nijhof** hat sich vor 12 Jahren eine clevere Art überlegt, TO zu implementieren. Schließlich gibt's ja verschiedene Typen, die alle mit TO zugewiesen werden wollen (Integer, Fließkomma, Double, Strings, etc.). Und wenn man mit den Prefixen schon mal angefangen hat, will man auch mehr (+TO...). Sein Trick: DOES> compiliert neben dem ausführbaren Code noch einen Pointer auf eine Prefix-Liste, die dann nach dem Prefix durchsucht werden kann --- der entsprechend angepaßte Code zum jeweiligen Prefix wird dann auch gefunden.

Einen weiteren Trick verwendet er, um sich eine interaktive Version des jeweiligen TOs zu sparen. Schließlich sind die meisten TOs ja mit POSTPONE definiert. Der Trick besteht darin, einfach die compilierende Version in einen temporären Puffer compilieren zu lassen, und dann diesen Code auszuführen.

**Willem Ouwerkerk** berichtete über die "Arbeitsgruppe Ushi", die an einem mit Forth programmierten Roboter bastelt. Der Roboter soll nicht vom Tisch fallen, Hindernissen ausweichen können, zeichnen, einer Linie folgen und sonst noch ein paar Kinderspiele können. Programmiert wird er in ByteForth. Hard- und Software sind vorhanden, und eine Baubeschreibung wird auch (wenn sie fertig ist) zugänglich sein.

**Martin Bitter** hatte auf der letzten Tagung einen Wettbewerb veranstaltet, bei dem mit einem Lego-Roboter das Wort "FORTH-2001" auf ein Blatt Papier gemalt



werden mußte. Das ging damals mehr oder weniger gründlich schief. Die versammelten Experten waren zur Auffassung gekommen, daß die Aufgabe unabhängig von der gegebenen Zeit mit der vorhandenen Hardware nicht lösbar war. Martin Bitter bekam die Aufgabe, diese Einschätzung bis zur nächsten Tagung zu widerlegen.

Er hatte also zwei Lösungen mitgebracht. Die erste Lösung war ein "Cheat", das Wort "Forth-2001" lief als Laufschrift über das RCX-Display. Für die "richtige" Lösung gab er sich mehr Mühe. Zunächst bekam der Roboter noch Sensoren, um einen definierten Ausgangspunkt zu finden --- damit sich die Abweichungen im Zeichenprogramm nicht akkumulierten. Die dazu notwendigen Leisten hat Martin angeblich aus dem Tagungsraum in Hamminkeln entwendet, und die Sensoren lagen auch in den Lego-Kästen herum. Allerdings gelang es Martin dann nicht, den Roboter wirklich live das Wort "Forth-2001" auf das Papier schreiben zu lassen --- die bekannten Probleme waren einfach zu groß.

**Fred Behringer** setzte seinen Vortrag vom letzten Mal einfach fort: Diesmal kam die Konvertierung von DNF in die Multilinearform (MLF) dran. Dazu werden die Terme nach der Anzahl vorkommender Variablen (nicht negiert) sortiert. Die Koeffizienten der vorangehenden Terme sind schon bekannt, die folgenden Terme spielen keine Rolle, also kann man den aktuellen Term aus den bereits berechneten Koeffizienten und dem Resultat der DNF berechnen. Der Algorithmus, den Fred vorstellte, verwendet drei verschachtelte Schleifen, wobei das Publikum der Meinung war, zwei würden reichen, wenn man eine Funktion hat, die die von Fred benutzte Ordnung aufzählen kann.

Nach der Pause fand sich ein geschickter Bogen vom Multitasker zum Forth-Prozessor

**Klaus Zobawa** zeigte sein  $\mu$ Task-System. Statt "richtige" Tasks gibt's nur eine Liste, die von einem Scheduler abgearbeitet wird. Alle Aktionen, die in diese Liste eingehängt werden, müssen den Stack so zurückgeben, wie sie ihn bekommen haben. Für einfache zeitgesteuerte Aufgaben ist das völlig ausreichend, ebenso für Aktionen, die von einem Interrupt angestoßen werden.

**Klaus Schleisiek** verband geschickt das Thema Multitasking mit seinem MicroCore, einem Forth-Prozessor im FPGA. Was macht man denn, wenn man einen IO-Port abfragt, dessen Ergebnis noch nicht bereit ist? Normalerweise bräuchte man noch ein Flag und müßte dann abhängig von diesem Flag *pause* aufrufen. Klaus legt einfach eine weitere Leitung in seinen Core, die signalisiert, daß die Peripherie noch nicht so weit ist, und damit einen Interrupt (oder eine Exception) auslöst. Die dabei angesprungene Routine kann den Task wechseln und bei der Rückkehr wird einfach der Befehl nochmal ausgeführt --- irgendwann liegt ein Wert am IO-Port an, und dann geht's weiter.

**Bernd Paysan** stellte mit dem *b16* seine Version vom Forth-Prozessor im FPGA vor. Der *b16* lehnt sich recht eng an Chuck Moore's jüngste Forth-Prozessoren an. Bernd präsentierte den Code (nur ein paar Hundert Zeilen), ein paar Code-Beispiele, und zeigte kurz die Entwicklungsumgebung, die mit einem von Hans Eckes gelöteten Board läuft. Der *b16* ist in Verilog geschrieben, läßt sich für FPGAs und ASIC-Prozesse synthetisieren, und ist unter GPL frei verfügbar.

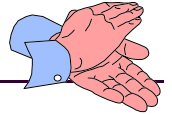
Am Nachmittag besuchten wir das Franz-Marc-Museum. In Oberbayern gibt es nämlich nicht nur traditionelle "Lüftl-Malerei" und kitschige König-Ludwig-Schlösser, sondern auch moderne Kunst. Nach diesem Abstecher in Kunst und Kultur ging's mit

**Jens Wilke** und "Instant" weiter. "Instant" ist eine Erweiterung des Gforth-Cross-Compilers, der es ermöglicht, richtige Forth-Programme mit diesem zu compilieren. Ein "richtiges" Forth-Programm führt gerade eben definierte Wörter im Lauf der Compilation aus. Das geht erstaunlicherweise ohne eine Emulator oder Simulator, vorausgesetzt, die Primitives des cross-compilierten Forths entsprechen denen des compilierenden Forths. Dann braucht man nur die gefundenen Wörter auch auf dem Host zu compilieren, und kann diese dann gleich ausführen. Speicherzugriffe werden über denselben Mechanismus abgearbeitet, der auch zur Initialisierung von Variablen verwendet wird; landen also im Image. *see* und der Debugger funktionieren auch (da es sich ja um ganz normalen Code handelt). Nur die Handhabung der Erweiterung kann noch etwas verbessert werden. Jens brachte es auf die Schnelle nicht fertig, das alles auch zu demonstrieren.

**Thomas Beierlein** regte an, sich mehr um TCP/IP zu kümmern. Es reicht nicht, dies oder jenes High-Level-Protokoll (wie einen Web-Server) zu implementieren, auch die darunterliegenden Schichten müssen her. Der Stand der Dinge ist für kleine Prozessoren nicht sehr erfreulich. Es gibt zwar den BSD-Stack als Referenz, der paßt aber nicht 'rein. Die anderen Stacks sind nicht in Forth geschrieben, und haben reichlich Bugs. Howard Oakford hat schon einen PPP-Stack geschrieben, und ist dabei, TCP zu implementieren, ist aber noch nicht fertig.

**Ulrich Hoffmann** ist dabei, VDs nach PDF zu wandeln. Bei den neueren Ausgaben ist das auch recht schnell gegangen, da die alle noch in elektronischer Form vorhanden sind, und per Acrobat ruck-zuck in PDF gewandelt werden können. Die älteren Ausgaben müssen erst eingescannt werden, was bei etwa 1000 Seiten doch recht viel Arbeit ist.

*Bernd Paysan*



## Stimmungsbild von der Forth-Tagung 2002

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

Die Tagung in Garmisch-Partenkirchen war sehr lebendig. Viele Mitglieder hatten Martin Bitter ihr Vertrauen für den freierwerbenden Direktorsposten angetragen. Nach zweimaligem Kopf-an-Kopf-Rennen konnte schließlich Bernd Paysan die meisten Stimmen auf sich vereinen. Wir freuen uns alle, mit Bernd einen international bekannten und anerkannten Forth-Fachmann, der noch dazu die jüngere Generation vertritt, in die Verantwortung für unseren Verein nehmen zu dürfen.

Der Trubel war so groß, daß wir es ganz und gar versäumten, dem langjährigen Direktor der Forth-Gesellschaft, Professor Dr. Thomas Beierlein, für seine gewiß nicht immer leichte Arbeit im Dienste der Gemeinschaft zu danken. Ich spreche ganz bestimmt im Namen aller, wenn ich das hiermit an dieser Stelle nachhole.

Mit Hans Eckes wurde (nach einigen Beschwörungsformeln im Drachenrat - der Drache nickte mit den Köpfen) ganz sicher ein würdiger Hüter für das Jahr 2002 gefunden, der zudem ebenfalls der Jugend ein angemessenes stärkeres Gewicht verleiht.

Fritz Prinz, der ja die Redakteuraufgaben wieder übernommen hat, schrieb mir, kaum dem Heimreiseauto entstieg, in einer enthusiastischen E-Mail, das sei eine der besten Tagungen gewesen, die er erlebt habe. Nur "eine der besten"? An Heinz Schnitter gewandt, darf ich sagen: Die beste der besten! Heinz hat sich damit aber selber eine Falle gestellt: An Forth-Ruhe im Ruhestand wird wohl nicht mehr zu denken sein. Bis in alle Ewigkeit (also bis zum Ende der Forth-Gesellschaft) wird er dazu verdammt sein, mit Rat und Tat zu helfen.

Heinzens Organisationstalent ist zu bewundern. Die Lebendigkeit der Tagung 2002 ist aber nicht zuletzt auch der ausnehmend guten Zusammensetzung der Tagungsteilnehmer zu danken. Unmengen von aktiven Mitgliedern waren vertreten! (Von wegen, Forth liegt im Sterben!) Die Generationen waren gut durchmischt und der starke Frauenanteil fiel wirklich angenehm ins Bild. (In vorgerückter Stunde wurde beim süffigen Bier auch über Nachwuchsfragen nachgedacht - über den Nachwuchs in Forth oder den Forthbestand des Nachwuchses oder so ähnlich.)

Die für die Beschlußfähigkeit benötigte Anzahl von anwesenden Mitgliedern wurde in der Versammlung am Sonntag diesmal mehr als zweimal überschritten. Verschiedentlich wurde über die doch recht vielen "schlafenden Mitglieder" gesprochen, "schlafend" im Sinne von "schlafender Energie". Überraschend sahen wir diesmal auch Filippo Sala bei uns, ein "Uraltmitglied", aber quicklebendig und forth-begeistert, alles andere als ein "Schläfer". Er hatte halt erst erweckt werden müssen. (Forth ist ganz bestimmt nicht alles, zum Verein gehört mehr.) Er hat mir

versprochen, in seinem Heimatland Italien nach versteckten Forth-Aktivitäten Ausschau zu halten, mit denen wir Verbindung aufnehmen könnten.

Das Ausland war gut vertreten. Dr. habil. Anton Ertl aus Österreich gehört ja fast zu uns - zum deutschen Sprachraum. Und er hat einen sehr interessanten Vortrag zur Code-optimierung gehalten. Ganz besonders hat mich als Beauftragter für Auslandsarbeit gefreut, daß Willem Ouwerkerk, der Vorsitzende der HCC Forth-gebruikersgroep, und Albert Nijhof, der Redakteur des Vijgeblaadjes, unserer Einladung gefolgt waren und mit ihren Vorträgen und Roboter-Exponaten sehr zum Gelingen der Tagung und zur Vielfalt des Erscheinungsbildes beigetragen haben. Es hat viele Gespräche mit ihnen gegeben und ich bin sicher, daß sich das sehr fruchtbringend auf den Ausbau der Zusammenarbeit mit unseren niederländischen Forth-Freunden auswirken wird. Martin Bitter, bekannt für seine spontanen Entschlüsse, hat sich vorgenommen, bei der nächsten "Bijeenkomst" in Maarssebroek (wo auch immer das liegen mag) mal kurz vorbeizuschauen und sein Holländisch aufzufrischen. Sprachlich gab es im übrigen mit unseren Gästen nicht die geringsten Probleme. Sie sind beide mehrsprachig orientiert.

Der Besuch im Walchenseekraftwerk kam einem mit den technischen Erläuterungen der Führung eigentlich recht vertraut vor, bei Franz-Marc im Museum wurde einem aber schnell klar, wie sehr man im einseitig ausgerichteten Berufsleben die allgemeine Bildung vernachlässigt hat. Wirklich sehr interessant. Merke: Außerhalb von Forth gibt es auch noch ein Leben. Man müßte es vielleicht in Forth einbeziehen!

Ewald Rieger war mit seiner liebenswerten Gattin schon mit dem Gedanken zur Tagung gefahren, die Tagung 2003 auszurichten. Der allgemeine Drang nach Aktivität war so groß, daß ihm Ulli Hoffmann beinahe den Rang als Tagungsausrichter in spe abgelaufen hätte.

Es rechnet also jeder damit, daß es Forth und dessen Gesellschaft auch im Jubiläumsjahr 2004 noch geben wird - und mir macht das alles riesigen Spaß. Es geht nicht nur um Forth, es geht in allererster Linie auch um den Verein.

*Fred Behringer*



## Die Legende lebt...

### Bericht aus dem Drachenrat!

Zwar muss jeder Drachenhüter ein feierliches Gelübde ablegen, Stillschweigen zu bewahren über alles das, was in der Drachenrunde gesprochen wird, aber über die Umstände und die Besonderheiten, die dort erkennbar sind, will und darf ich berichten.

Bei gedämpftem Licht scharrt sich ein kleines Grüppchen ehemaliger Drachenhüter um das Abbild des einen SWAP zusammen. Ein wenig verloren wirkt es schon in dem fast saalartig großen Raum. Die Stimmung ist getragen, mystisch erwartungsvoll. Wen der Anwesenden wird SWAP mit Erleuchtung füllen, den zukünftigen Drachenhüter mitzuteilen? Denn das ist gewiss: SWAP erwählt sich seinen Hüter selbst! Die Ehemaligen sind 'nur' Medien seines Willens.

Andererseits: der Geist eines jeden Drachenhüters ist gestählt durch die enge Verbundenheit des gemeinsam verbrachten Jahres.

So geschieht es nicht selten, dass SWAPs Eingebungen und die Gedanken der Hüter nicht übereinstimmen – und je mehr Hüter, desto mehr abweichende Gedanken.

Dann verschwimmt SWAPs Bild, die kleine Statue scheint unscharf zu werden und ein sichtbarer Nebel umhüllt den Geheimnisvollen. Dieser Nebel schwillt an, wird größer und mächtiger und verhüllt die Häupter der Hüter. Eine klare Sicht ist nicht möglich. Ich fühle mich der Welt entrückt.

Drachenwasser wird gereicht – der Nebel bleibt. SWAP ist mächtig aber gerecht. Er versucht nicht den Eigensinn der Hüter zu brechen, er argumentiert auf eine mir unerklärliche Weise mit mir. Ich höre fast seine Gedanken in meinem Kopf.

Langsam schält sich aus dem Nebel das Bild eines belebten SWAPs! Er erhebt sich, mit ruhigem, majestätischen Flügelschlag und zieht unter der holzgetäfelten Decke einen weiten Kreis. Das ist 'mein' SWAP! Nicht das er mir gehört – nein das nicht. SWAP gehört sich selbst. Aber, dieser SWAP ist nur für mich da! Seine Gedanken höre ich, seine Wünsche fühle ich.

Schemenhaft erkenne ich einen weiteren SWAP. Und noch einen, und noch einen. Ich zähle genauso viele Inkarnationen von SWAP, wie Hüter um den Tisch versammelt sind.

Jeder SWAP reiht sich ein in einen Kreis und zieht, einer nach dem anderen, seine Runden. Kein Laut ist zu hören, nur dann und wann ist das Ächzen eines Drachenhüters zu vernehmen. Langsam bekommt SWAP (oder die SWAPs?) Farbe. Er wird grüner und grüner, ja er beginnt geheimnisvoll zu leuchten und zu strahlen – heller und heller.

Gleißend wird sein Schein, von Nebel ist nichts mehr zu sehen. Seine Argumente dröhnen in meinem Geist – mein Schädel dröhnt.

Stundenlang, so scheint es, geht dieses stumme Ringen von sich. Dann mit einem Schlag, senken sich alle SWAPs einander durchdringend auf die kleine Statuette und versinken in ihr.

Erschöpft blicke ich auf. In den Augen aller anderen Hüter sehe ich die Gewissheit, den wahren, neuen Hüter zu kennen. Es ist entschieden!

*PS Einen kleinen Eindruck, des multiplen SWAPs kann man unter <http://www.jwdt.com/~paysan/bigforth.html> bekommen, wenn man BigForth installiert und dragon-shoot.m aufruft.*

*Martin Bitter*

Hallo Friederich,

ich habe versucht meinen Forthweg, den ich bisher beschritt, niederzuschreiben. Um die Leute nicht zu langweilen konzentrierte ich mich auf das Wesentliche.

#### Mein Weg zum Web Anwendungsserver

Eigentlich wollte ich als nächstes einen Artikel darüber schreiben, warum es in Forth besonders einfach ist, Business Anwendungen zu entwickeln.

Leider kamen (und kommt) mir meine letzten 10 Jahre C, Smalltalk und Java KnowHow in die Quere.

Dazu muss man allerdings wissen wie alles begann. Nach den ersten OO Schritten, kam im Laufe der Jahre ein immer besseres Verständnis. Ich schrieb immer besseres OO, bis eines Tages wirklich nur noch OO – und sonst nix – da war. Aber etwas Schreckliches war passiert. Je mehr ich OO betrieb, um so umfangreicher und undurchsichtiger wurden die Programme. Die Klassen wurden immer mehr, die Methoden kämpften darum abgeleitet zu werden bis ... es mir reichte.

Es wurde mir einfach zu bunt. Ein nicht unwesentliches Merkmal der Wartbarkeit und damit auch der Stabilität der Anwendung, ist, neben einem klaren, einfachen und logischem Programmfluß, dessen Größe. Punkt.

Also verfiel ich nun in das andere Extrem - Minimalismus. Geprägt durch Forth, Chuck Moore und viele wertvolle Beiträge der Forthgemeinde, begann ich einen neuen Weg. Und der war und ist steinig. Wie Moore wollte ich ab nun kein Gramm Forthfett zuviel. Jedes Wort muss seine Anwesenheit begründen. Da war bzw. bin ich in einem ziemlichen Dilemma. Ich kannte und konnte (bzw. kann immer noch) zu wenig Forth um zu entscheiden, was ich wie lösen sollte. Meine eigene Erfahrung ist mir zum Verhängnis geworden.

Da war noch das Problem, mit welchem Forth ich denn das nun alles anstellen sollte. Die Auswahl ist schier unendlich. Und außerdem wollte ich mein eigenes Forth auch noch bauen (na klar, wer nicht :-). Also begann ich, nach einem Forth zu suchen das `_alles_` kann. Die waren mir aber viel zu groß. Ich verstand zu wenig.



Also habe ich mit sämtlichen Forth's experimentiert. Commodore64, 128, AtariST, DOS, Windows und Linux boten eine breite Spielwiese.

An dieser Stelle möchte ich Bernd Paysan besonders für sein BigForth für AtariST, das er mir großzügiger Weise komplett überlassen hat, danken. Ja, auf dem AtariST habe ich meine ersten Programme und Erfahrungen mit Forth gemacht. Diskettenlaufwerk, 1MB Ram, 8MHz 68kCPU und keinerlei Wartezeiten - ein Wahnsinns-Gefühl :-)

Leider muss ich meine Produktion auf diversen Unixen laufen lassen (HPUX, Solaris, Linux, ...). Daher blieben mir zwei Alternativen: ein eigenes Forth oder gForth. Anton Ertl, Bernd Paysan und andere haben dieses prachtvolle gForth entwickelt. Aber es widerstrebte meinem Drang zum Minimalismus und daher baute ich doch ein eigenes minimal Forth (mit den Ideen des direct threadings aus gForth). Ich konnte eine VM mit sämtlichen Primitiven und C-Callout in 1000 Zeilen unterbringen, was meiner Meinung nach kein schlechter Wert für ein C Programm ist, das Compiler, Interpreter und VM in sich birgt :-)

Der C Code (gcc spezifisch) ist knapp 10 mal schneller als eine reine ANSI-C Implementation. Und damit ich nicht wieder in Komplexität versinke, installierte ich mir einen 386/33 als Entwicklungsserver (mein 386/16 ging leider kaputt). Das Einzige was auf der Kiste echt lahm ist, ist der gcc. Der braucht für meine 1000 Zeilen C Code knapp 2 Minuten. Der Compiliervorgang des Forthcodes ist nicht merkbar.

Jeder der meint, dass man, nachdem man einen Forthcompiler gebaut hat, nicht Forth kann, hat unbestritten recht.

Und da stehe ich nun. Ein Jahr nach meinen ersten Anfängen stehe ich in der Entwicklung eines Web-Anwendungsservers. Ich komme mir wie ein kleines Kind vor, das noch über seine eigenen Beinchen stolpert. Es fällt mir sehr schwer 'einfach' zu denken und die 'nicht Probleme' in Ruhe zu lassen. Ein dynamisches Objektnetz mit Garbage Collection und dynamischem Polymorphismus in Forth zu bauen, ist absolut kein Problem. Das Problem ist, diesem Drang zu widerstehen und einfachere Lösungen zu suchen. Bis jetzt kennt und braucht mein Forth kein allocate, free etc. Und das wirklich Überraschende an der Sache ist, dass die Lösungen wirklich sehr sehr sehr viel einfacher werden. Man beginnt die Unzulänglichkeiten diverser Internetprotokolle zu erkennen, weil sie sich gegen eine klare Implementierung in Forth sträuben. Vielleicht sollten die RFC Herren mal ihr Zeug mit Forth implementieren, dann würden sie viele Schwachstellen sofort erkennen.

Ein, meiner Meinung nach, elegantes Forthprogramm fließt wie ein ruhiger Bach :-)

Ich bin noch am Lernen (wer hat gesagt Forth lernt man in einer Stunde?), und jeden Tag lerne ich Probleme noch einfacher und eleganter zu lösen, was unheimlich viel Spaß macht. Erkenne ich einen einfacheren Weg ein Problem zu lösen, reimplementiere ich den Teil einfach neu.

Dieser Aufwand ist leider nötig, um sich zu verbessern.

Meine Pläne für die Zukunft sind zunächst die Fertigstellung des Web Anwendungsservers. Elegante Lösungen für Datenbanken und HTML Workflow fehlen mir noch, Ideen habe ich jede Menge :-)

Danach möchte ich die Erfahrungen in Form von Forth Konzepten (Patterns) veröffentlichen. Wobei ich die Ideen in den Vordergrund stellen möchte; wie z.B. der Einsatz des Dictionaries, des Returnstacks, aber auch String Handling und Parsing. Vielleicht gelingt es mir, damit mehr Begeisterung für Forth in der Nicht-Forthwelt zu erlangen.

Andreas Klimas  
www.klimas-consulting.com

---

*Ich hatte Andreas Klima um einen kleinen Beitrag für diese Ausgabe der VD gebeten. Die Antwort, und mit ihr der Erfahrungsbericht, den Sie gerade gelesen haben, kamen umgehend. Vielen Dank, Andreas Klima. Vielen Dank für die Arbeit, die auch in einem solchen Bericht steckt, und vielen Dank für die interessanten, philosophischen Ansätze !*

### **Ein ... elegantes Forthprogramm fließt wie ein ruhiger Bach :-)**

*Von dem was wir Forther seit rund 25 Jahren anderen Menschen nahe zu bringen versuchen, steckt in diesem Satz so Vieles, daß ich ihn einfach wiederholen mußte. Einfachheit, Klarheit, überschaubare Strukturen deren Komplexität sich an dem gegebenen Problem orientiert sind die Grundlage von Schönheit, Freude und der Gewißheit, etwas vollbracht zu haben.*

*Jetzt freue ich mich auf die „Patterns“, die wir hoffentlich auch für die VD bekommen.*

fep



Das neue Konzept:

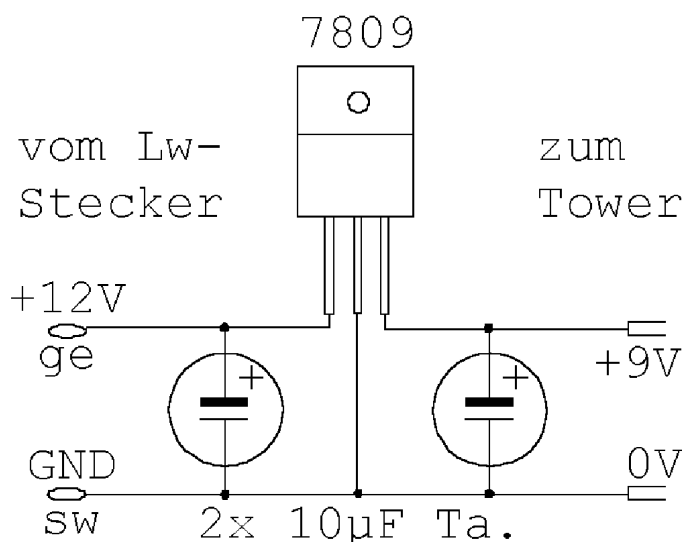
virtueller Kaffee,  
volldigitalisiert,  
geschmacksneutral...



## Tower-Power

Rolf Schöne

Der LEGO IR-Turm ist über ein seriell Nullmodem-Kabel dauernd mit dem PC verbunden. Es bietet sich daher an, die Batterie aus dem Turm zu entfernen und ihn aus dem PC zu versorgen. Dazu sind die im PC an einem Laufwerkstecker abgreifbaren +12V auf +9V zu reduzieren, was mit der folgenden kleinen Schaltung geschehen kann, deren +9V-Ausgang mit dem +9V-Anschluß auf der Turm-Platine zu verbinden ist:



*Für die Art der Ausführung sind keinerlei Grenzen gesetzt. Man kann das fliegend verdrahten, auf eine Lochrasterplatine löten, die Schaltung unterwegs offen liegen lassen, in eine Schachtel einbauen, in den PC schrauben, auf einem Slotblech montieren, im geräumigen Batterieschacht des Towers oder im Tower selbst unterbringen, ... Sie haben sicher noch andere, womöglich bessere Ideen.*

Sollten Sie wie ich das COM-Port ausschließlich für den Tower verwenden, so gibt es noch eine weitere saubere Lösung, welche den zusätzlichen +9V-Draht zum Tower einspart:

Das mitgelieferte serielle Kabel ist nicht voll belegt. Die Pins 1 (DCD), 6 (DSR) und 9 (RI) sind frei, allesamt Eingänge zum PC, und können gefahrlos mit +9V beaufschlagt werden. Bringen Sie die kleine Schaltung im PC unter, löten Sie an einen der Pins 1, 6 oder 9 die +9V an (notfalls über einen Adapter, wenn Sie auf ATX-Boards nicht so gerne herumlöten), machen Sie ein neues Kabel (s. VD 04/2001 S. 10), bei dem auch der gewählte Pin 1:1 verdrahtet ist, und verbinden Sie den gewählten Pin auf der Tower-Platine mit dem +9V-Anschluß.

Als 7809 ist jeder beliebige Typ verwendbar. Ich habe den L7809CV von ST im Gehäuse TO-220 verwendet, der 1A liefern kann. Der Tower benötigt nur 10mA. Eine Kühlung ist daher nicht erforderlich. Als Kondensatoren empfehlen sich Tantal-Elkos mit mindestens 4,7µF.

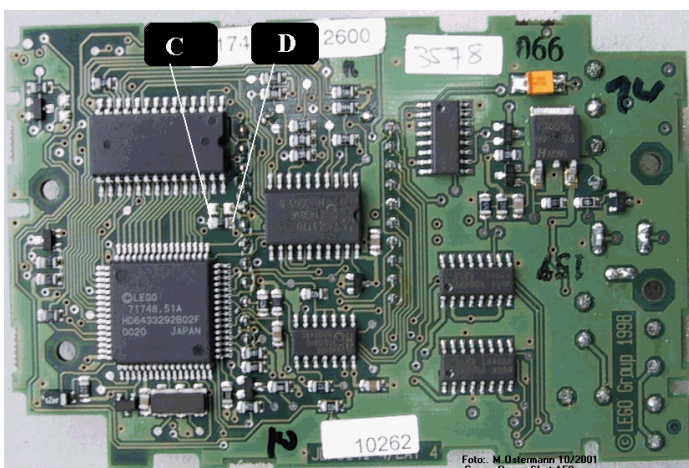
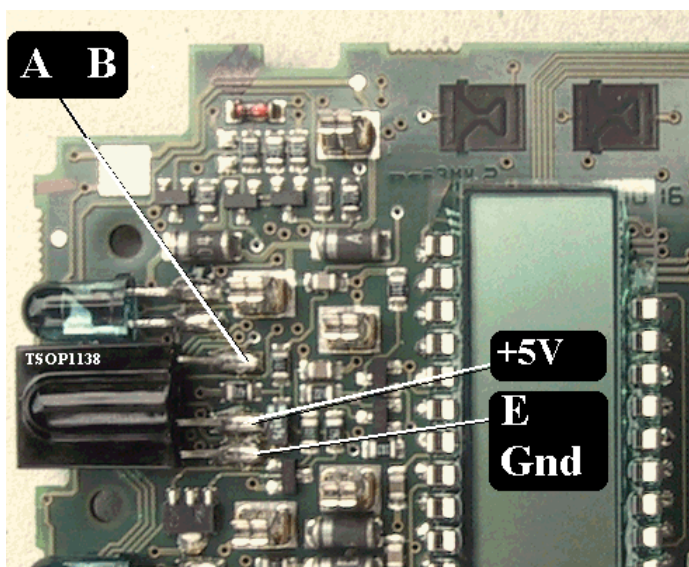
Fragen bitte an [rolf@rolf-schoene.de](mailto:rolf@rolf-schoene.de)

## Vorschau auf kommende Beiträge zum RCX

Michael Kalus und Adolf Krüger werden berichten:

### RCX am Draht

*Klar ist uns wieder geworden, daß eine gute und einfache Verbindung mit rein passiven Bauteilen nicht möglich ist: Die RS232 verwendet inverse Logik, aktive Bits sind LOW, auch am H8. Auch muß eine Pegelumsetzung von 0/+5V auf bis zu -12/+12V gemacht werden – so weit also das klassische Problem. In Frage kommt die Verbindung mittels Optokoppler (TLP621-2) oder mit Hilfe eines seriellen Schnittstellen Bausteins (MAX232).*



## Forth-Gruppen regional

- Moers**      **Friederich Prinz**  
Tel.: **02841-58398** (p) (Q)  
(Bitte den Anrufbeantworter nutzen !)  
**(Besucher: Bitte anmelden !)**  
Treffen: 2. und 4. Samstag im Monat  
14:00 Uhr, **MALZ, Donaustraße 1**  
**47441 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-Neuostheim**
- München**      **Jens Wilke**  
Tel.: **089-89 76 890**  
Treffen: jeden 4. Mittwoch im Monat  
**Ristorante Pizzeria Gran Sasso**  
**Ebenauer Str. 1**  
**80637 München**

## µP-Controller Verleih

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

## Gruppengründungen, Kontakte

**hier könnten SIE  
sich zur Gründung einer  
lokalen Gruppe zur Verfügung stellen !**

## Forth-Hilfe für Ratsuchende

### **Forth allgemein**

Jörg Plewe  
Tel.: 0208-49 70 68 (p)

Jörg Staben  
Tel.: 02103-24 06 09 (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  
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



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.

2 / 2002

Photos: Ulrich Hoffmann  
[www.forth-ev.de/photos/forth-2002](http://www.forth-ev.de/photos/forth-2002)

