

Forth Magazin
Vierte Dimension
Volume I/Nr. 3+4
Preis 5,00 DM

VIERTE DIMENSION

DFÜ

Networking
Forth Processoren
Industrie Systeme
C 64 FORTH'S
Gruppen

EDITORIAL

Liebe Forthfreunde,

nun ist es doch noch gelungen, das Forth Magazin, die VIERTE DIMENSION, herauszubringen. Dazu als allererstes meinen allerherzlichsten Dank an Dieter Hilbig und Michael Kalus, die noch am zweiten Advent ihre Familien im Stich gelassen haben, um das Layout zu machen. Und an Angelika Theis, die die Texte in den Kaypro gebracht hat.

(Und den Leuten in Hamburg drücke ich die Daumen, das sie es schaffen, den Druck und Versand vor Weihnachten fertig zu stellen und das Blatt zu den Forthlern unter den Gabentisch zu bringen. MK)

Wie kam es zu der langen Pause, in der keine Ausgabe zustande gekommen ist? Tja, Leute, da haben wir unsere Kräfte bei der Forth Gesellschaft doch überschätzt. Vorrangige hauptberufliche Arbeit hat verhindert, daß die doch weitgehend ehrenamtliche Arbeit an so einer Zeitung vorankam. Es mußten erst zusätzliche Menschen gefunden werden, die dieses machen wollten. Nun war die Jahreshauptversammlung im November in Bergzabern, und bei dem Treffen dort hat sich erfreulicherweise der Knoten gelöst. Der Zusammenbau der VIERTEN DIMENSION konnte nach Wuppertal in die dortige lokale Gruppe verlagert werden. So ist diese Ausgabe also aus dem Material, daß in Hamburg bei der Forth Gesellschaft eingegangen ist, dort zu diesem Doppelheft zusammengebaut worden. So wollen wir es auch in Zukunft halten. Das Erscheinungsbild mußte deswegen nocheinmal überdacht werden. Um das Verfahren reibungsloser und kostengünstiger zu machen, wurde auf den Spaltaufbau per Lichtsatz wieder verzichtet. Da es nun in diesem Jahr noch einmal schnell gehen sollte, konnte keine rechte Anzeigenredaktion mehr erfolgen. Das soll ab 1986 aber wieder gehen. Interessenten sollen sich weiterhin bitte an die FG wenden. Sowie so sendet bitte alles Material an die FG. Hier wird alles gesammelt, hier laufen die Fäden zusammen.

Ich wünsche Euch allen ein frohes Weihnachtsfest 1985 und einen guten Rutsch ins neue Jahr. Möge die Forth Gesellschaft in ihrem zweiten Jahr weiterhin so erfolgreich sein.

Euer H.G.Lynsche

Impressum:

Herausgeber: Forth Gesellschaft eV.
Chef v. Dienst: Horst-Günter Lynsche
Schanzenstr.27
2000 Hamburg 6
tel: 040-435070
Redaktion: Michael Kalus
Präsidentenstr.40
5830 Schwelm
tel: 02336-82204
Druck: Eckart Schmidt (Kodak)
Forth: Klaus Schleisiek
sowie alle namentlich genannten Autoren.

Auflage: 750 Exemplare in Europa

Erscheinungsweise: 4 Ausgaben/Jahr © Forth Gesellschaft eV 1985
Nachdruck nur auszugsweise und mit genauer Quellenangabe. Namentlich gekennzeichnete Artikel geben nicht unbedingt die Meinung der Redaktion wieder. Freie Mitarbeiter sind erwünscht, werden jedoch gebeten nur solche Artikel & Programme einzusenden, für die das Copyright bei Ihnen liegt. Wenn nicht anders angegeben, gehen veröffentlichte Programme in die Public Domain über. Manuskripte können als Diskette, per DFÜ (ASCII, 300 Bd) und als Schreibmaschinen - Texte in die Redaktion gesendet werden.

Es ist soweit !

Das ultraFORTH83 für den C64 ist fertig. Das Handbuch ist im Druck. Es umfasst 200 Seiten Din A5 mit vielen Zeichnungen. Die Quelltexte sind ebenfalls fertig. Sie liegen auf 2 Disketten, beidseitig bespielt, vor.

Anlässlich dieses freudigen Ereignisses haben wir uns entschlossen, ein Super-Sonder-Jubel-Angebot zu präsentieren:

Das ultraFORTH83 kostet:

45,- DM
für Forth-Gesellschaftsmitglieder

65,- DM
für Nicht-Mitglieder

100,- DM
für Nicht-Mitglieder incl. einem Jahr Mitgliedschaft in der Forth Gesellschaft e.V.

76,- DM
statt 100,- DM für Schüler, Studenten und Arbeitslose, sofern ein Beleg geschickt wird.

(Die Preise müssen wir später, wenn wir wissen, was die Herstellung wirklich kostet, bestimmt noch ändern.)

Überweist einfach die Summe (Adresse nicht vergessen !) auf das folgende Konto:

Bernd Pennemann
Sonderkonto U
Kto.Nr. 31787 - 204
PschA Hmb Blz 200 100 20

Wir schicken dann, so schnell wir können, ein Exemplar des ultraFORTH83.

Wir hoffen auf viele Bestellungen und viel Post !

ultraFORTH 83

MODEM7, CHRISTENSEN

von Edmund Ramm

Bei der Fernübertragung beliebiger, also auch binärer Daten leistet das sich in der Public Domain befindende Christensen-Protokoll, eingebaut in das MODEM7 DFÜ- und Terminal-Programm, hervorragende Dienste. Für FORTH-Enthusiasten bedeutet die Anwendung von MODEM7 jedoch ein Verlassen der vertrauten Programmierumgebung und könnte Entzugserscheinungen zur Folge haben.

Bereits 1983 erschien eine Veröffentlichung (1) von SEND und RCV geschrieben in PC/FORTH unter MS-DOS, und damals wurde das einfachere XMODEM-Protokoll (auch von Ward Christensen) implementiert.

Für Z80 fig-FORTH 1.1 (ab User Version g) unter CP/M 2.2 wurde (1) entsprechend modifiziert und gleichzeitig der auch in MODEM7 zur Anwendung gelangende CRC-16 ($x^{16} + x^{12} + x^5 + 1$) aufgrund seiner besseren Fehlererkennung eingesetzt. (100% aller Ein-, Zwei- und ungeraden Bitfehler, 100% aller Fehlerbursts ≤ 16 Bits, 99.9969% aller 17-Bit-Fehlerbursts sowie 99.9984% aller längeren Fehlerbursts (2). Eine 8-Bit-Prüfsumme gewährleistet dagegen nur eine Trefferrate von ca. 95% (3).)

Die Datenübertragung unter MODEM7 wird, wie auch bei XMODEM, von der empfangenden Seite angestoßen, es wird hier jedoch kein NAK vom Sender erwartet, sondern eine CRC-Anforderung in Form eines ASCII 'C', 43H. Der Empfänger gibt solange 43H aus, bis er entweder vom Sender SOH erhält oder am Terminal ^X (CAN) eingegeben wird.

Nach SOH übergibt der Sender zunächst die Nummer des zum Transfer anstehenden Sektors modulo 256 gefolgt vom 1er-Komplement selbiger Sektornummer. Der erste Sektor hat die Nr. 1, auf Sektor 255 folgt Sektor 0.

Im Anschluß an den Header werden die 128 Bytes eines Sektors bar jeder Kodierung übertragen.

Den Abschluß der Übertragung eines Sektors bildet der CRC-16, dessen höherwertiges Byte zuerst gesendet wird. Der Empfänger vergleicht den von ihm selbst errechneten mit dem empfangenen CRC-Wert und quittiert bei korrektem Erhalt der Daten mit ACK, widrigenfalls mit NAK. Erhält der Sender kein ACK, so wird der eben gesendete Sektor, eingepackt in SOH, Header und CRC-16, nochmals übertragen.

Wenn ACK erkannt wurde, wiederholt sich der o.g. Vorgang für den nächsten Sektor.

MODEM7, CHRISTENSEN

von Edmund Ramm

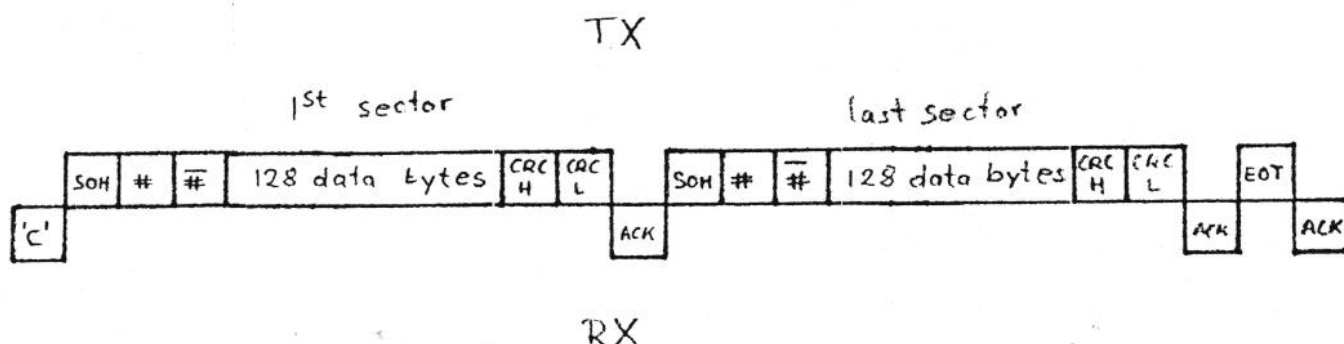
Nach der Übertragung des letzten Sektors der Datei und einer ordnungsgemäßen Quittierung schickt der Sender EOT über die Leitung, und der Empfänger beantwortet dies mit einem weiteren ACK.

Einzige Schwachstelle dieses Protokolls ist m.E. die Quittierung in Form eines einzelnen Bytes. Frage an die Statistiker unter uns: wie groß (oder hoffentlich klein) ist die Wahrscheinlichkeit, daß NAK durch eine Störung zu ACK verfälscht wird? Jedenfalls hätte an dieser Stelle eine Absicherung, z.B. ähnlich der für die Sektornummer, nicht geschadet. Andererseits ist dem Verfasser bisher keine Datei, selbst bei stark gestörter Verbindung, fehlerhaft übertragen worden. Da geschieht schon eher ein Abbruch des Transfers nach Überschreitung der maximal zugelassenen Timeouts.

Soweit unter CP/M 2.2 möglich, wurden Robert Taylors SEND- und RCV-Definitionen, sofern sie nicht von der Umstellung auf CRC-16 betroffen waren, unverändert übernommen. Die Systemmeldungen wurden an einigen Stellen etwas MODEM7-ähnlicher gestaltet. Abweichend von MODEM7 werden bei Abbruch der Übertragung die bis dahin empfangenen Daten nicht gelöscht. Bei Textdateien ist dies zeitweise recht nützlich; ein Programmfragment ist in der Mehrzahl der Fälle wertlos und kann durch XFER IN-NAME, Eingabe des Dateinamens und XFER ERA-FILE aus dem Directory gestrichen werden.

Hinzugefügt wurden die Definitionen TERM und ECHO, um einen beidseitigen Terminalbetrieb zu ermöglichen. Mit ^E werden beide Terminalmodi wieder verlassen.

Die Schnittstellen-Initialisierung und die UART-Treiber (SCR # 27) sind hardwareabhängig und bedürfen einer Anpassung an die jeweilige Konfiguration. Die hier gezeigten Definitionen gelten für eine Z80-CTC/Z80-SIO-Kombination mit 6.14MHz Takt.



SCR # 18

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

SCR # 19

0 (SEND & RCV files using MODEM7 protocol EHR 850420)
1 -->
2
3 Red face time: a major part of the following code has been
4 shamelessly pinched from Robert Taylor's "SEND and RCV" which
5 appeared in DDJ 09/83 p.66-81.
6
7 The routines presented here are not to be used for commercial
8 purposes, but unlimited personal, non-commercial use is allowed.
9
10
11
12
13
14
15

SCR # 20

0 (SEND & RCV files using MODEM7 protocol EHR 850420)
1 FORTH DEFINITIONS HEX FORGET TASK : TASK ;
2
3 CODE CRC-16 (u1 8b --- u2, X^16 + X^12 + X^5 + 1)
4 EXX B POP H POP 1021 D LXI C A MOV B B MVI
5 BEGIN
6 A C MOV 80 ANI H XRA A H MOV H DAD
7 CS IF
8 E A MOV L XRA A L MOV D A MOV H XRA A H MOV
9 ENDIF
10 C A MOV RAL B DCR 0=
11 UNTIL
12 H PUSH EXX PCIX
13 C;
14 DECIMAL
15 -->

Edmund Ramm, P.O.Box 38, D-2358 Kaltenkirchen, W.Germany

```

SCR # 21
Ø ( SEND & RCV files using MODEM7 protocol           EHR 85Ø42Ø)
1
2 DECIMAL
3
4 : <> = Ø= ;           : 4DROP  DROP DROP DROP DROP ;
5
6 : D= ROT = ROT ROT = AND ;   ( d1 d2 --- f)
7
8 : EXIT  R> DROP ;
9
1Ø : RESTART
11 ?COMP ?STACK Ø Ø >R >R
12 BEGIN DUP 1 <> WHILE >R >R REPEAT 2DUP [COMPILE] AGAIN
13 BEGIN R> R> 2DUP Ø Ø D= UNTIL 2DROP ; IMMEDIATE
14
15 -->

```

```

SCR # 22
Ø ( SEND & RCV files using MODEM7 protocol           EHR 85Ø42Ø)
1
2 : CREATE-FCB
3 <BUILDS HERE 165 ERASE 165 ALLOT DOES> ;   ( 37 + 128 bytes)
4
5 37 CONSTANT BUFFER-OFFSET   ( fcb length, i/o buf displacement)
6
7 : SET-DMA ( addr ---) BUFFER-OFFSET + 26 BDOS DROP ;
8
9 : ?BUFFER-ADDR ( fcb-addr ---) BUFFER-OFFSET + ;
1Ø
11 CREATE-FCB XFER           ( SEND & RCV work fcb )
12
13 -->
14
15

```

```

SCR # 23
Ø ( SEND & RCV files using MODEM7 protocol           EHR 85Ø42Ø)
1 ( fcb-addr --- dir-code)
2 : DCONIN 255 6 BDOS ;           : DCONOUT 6 BDOS DROP ;
3 : OPEN-FILE 15 BDOS ;           : CLOSE-FILE 16 BDOS ;
4 : ERA-FILE 19 BDOS ;
5 : MAKE-FILE 22 BDOS ;
6 : GET-SIZE 35 BDOS DROP ; ( fcb-addr ---; rØ r1 r2 in FCB)
7
8 ( fcb-addr --- Ø [error], buf-addr [success])
9 : READ-RAND DUP DUP SET-DMA 33 BDOS ( Ø = success)
1Ø IF ( <>Ø = error) DROP Ø
11 ELSE BUFFER-OFFSET + THEN ;
12 : WRITE-RAND DUP DUP SET-DMA 34 BDOS ( Ø = success)
13 IF ( <>Ø = error) DROP Ø
14 ELSE BUFFER-OFFSET + THEN ;
15 -->

```

Edmund Ramm, P.O.Box 38, D-2358 Kaltenkirchen, W.Germany

```

SCR # 24
0 ( SEND & RCV files using MODEM7 protocol           EHR 850420)
1
2 : GET_DRIVE      ( ^fcb[0] ^buf[0] --- ^fcb[1] ^buf[0] or ^buf[2] )
3 2DUP DUP 1+ C@ 58 = IF C@ 64 - SWAP C! 2+
4                      ELSE DROP 0 SWAP C! ENDIF
5  SWAP 1+ SWAP ;
6
7 : ?WILDCARD      ( n1 n2 n1 c --- n1 n2 n1 c f )
8  DUP 42 = OVER 63 = OR ;
9
10 : BEGIN_EXTENSION
11      ( ^fcb[n] ^buf[n] ^fcb[n] '.' --- ^fcb[9] ^buf[n+1] )
12  DROP DROP 1+ SWAP DROP XFER 9 + SWAP ;
13
14 -->
15

```

```

SCR # 25
0 ( SEND & RCV files using MODEM7 protocol           EHR 850420)
1
2 : BUF_FCB_C!
3      ( ^fcb[n] ^buf[n] ^fcb[n] c --- ^fcb[n+1] ^buf[n+1] )
4  SWAP C! 1+ SWAP 1+ SWAP ;
5
6 : BADNAME      ( --- )
7  ." Invalid file name" CR QUIT ;
8
9 : PARSE      ( ^fcb[0] ^buf[0] --- )
10 OVER 12 BLANKS ( space fill fcb-filename) GET_DRIVE
11 BEGIN 2DUP C@ DUP 0 <> OVER BL <> AND
12 WHILE ?WILDCARD IF 4DROP BADNAME ENDIF
13   DUP 46 = IF BEGIN_EXTENSION ELSE BUF_FCB_C! ENDIF
14   OVER XFER 12 + > IF DROP DROP BADNAME ENDIF
15 REPEAT 4DROP XFER 12 + 23 ERASE ; -->

```

```

SCR # 26
0 ( SEND & RCV files using MODEM7 protocol           EHR 850420)
1
2 : FILENAME      ( fcb-addr --- )
3  BL WORD HERE 1+ PARSE ;
4
5 : IN-NAME      ( fcb-addr --- )
6  ." Filename: "   HERE 15 EXPECT   HERE PARSE ;
7
8 : OUT-NAME      ( ^fcb[0] --- )
9  DUP C@ -DUP IF 64 + EMIT 58 EMIT ENDIF
10 DUP 1+ 8 TYPE 46 EMIT 9 + 3 TYPE ;
11
12 -->
13
14
15

```

Edmund Ramm, P.O.Box 38, D-2358 Kaltenkirchen, W.Germany

SCR # 27

```

0 ( SEND & RCV files using MODEM7 protocol          EHR 850420)
1 ( Hardware dependent, here for Z80B-CTC & Z80B-SIO)  HEX
2 : ?COM  E1 P@ 1 AND ;                               : ?TXRDY  E1 P@ 4 AND ;
3 : WAITOUT  BEGIN ?TXRDY  UNTIL ;
4 : WAITIN  BEGIN ?COM  UNTIL ;
5 : XIN      E0 P@ ;                                   : DOUT  E0 P! ;
6 : XOUT     1+ -DUP IF WAITOUT 1-  E0 P!  ENDIF ;
7 : CLEAR    ?COM IF XIN DROP ENDIF ;                 ( empty rx buffer)
8 : SETSIO   00 E1 P!  18 E1 P!  ( reset SIO A)
9           01 E1 P!  00 E1 P!  ( disable INT generation)
10          04 E1 P!  44 E1 P!  ( *16, 1 stop bit)
11          03 E1 P!  C1 E1 P!  ( 8 bits RX, RXenable)
12          05 E1 P!  EA E1 P! ; ( DTR, 8 bits TX, TXen, RTS)
13 : I300    0D E8 P!  50 E8 P! ; ( CTC 6.14MHz/16/16/Bdtrate)
14 DECIMAL
15 -->

```

SCR # 28

```

0 ( SEND & RCV files using MODEM7 protocol          EHR 850420)
1
2 0 VARIABLE CRC          0 VARIABLE ?ECHO
3 0 VARIABLE 1STRCV      0 VARIABLE 1STSEC
4 0 VARIABLE ERRS        10 CONSTANT MAXERR      0 VARIABLE NAK-CT
5
6 0 VARIABLE RSEC        0 VARIABLE SSEC          0 VARIABLE XSEC
7
8 0 CONSTANT NUL         24 CONSTANT CAN          1 CONSTANT SOH
9 4 CONSTANT EOT         6 CONSTANT ACK          21 CONSTANT NAK
10
11 0 VARIABLE TIMER      10000 VARIABLE TIM-CT
12 : LONG-TIMEOUT 30000 TIM-CT ! ;
13 : SHORT-TIMEOUT 10000 TIM-CT ! ;
14 -->
15

```

SCR # 29

```

0 ( SEND & RCV files using MODEM7 protocol          EHR 850420)
1
2 : RCV-SETUP  SETSIO I300
3 XFER FILENAME CR XFER OPEN-FILE 255 <>
4 IF ." File exists, delete? (Y/N) " KEY CR
5     DUP 89 <> SWAP 121 <> AND IF QUIT ENDIF
6     XFER ERA-FILE DROP
7 ENDIF
8 XFER MAKE-FILE 255 =
9 IF ." Can't create " XFER OUT-NAME CR QUIT ENDIF
10 XFER OUT-NAME ." open." CR
11 0 XFER 33 + ! 0 RSEC ! 0 XSEC ! 0 NAK-CT !
12 ;
13 -->
14
15

```

Edmund Ramm, P.O.Box 38, D-2358 Kaltenkirchen, W.Germany

```

SCR # 30
0 ( SEND & RCV files using MODEM7 protocol          EHR 850420)
1 : TIMED-IN    TIM-CT @ TIMER !
2 BEGIN        ?COM IF XIN EXIT ENDIF    -1 TIMER +!    TIMER @ 0=
3 UNTIL    -1 ;
4
5 : GOBBLE      BEGIN TIMED-IN -1 = UNTIL ;
6
7 : ABEND       CR ." Aborting."    XFER CLOSE-FILE 255 =
8              IF CR ." Can't close " XFER OUT-NAME  ENDIF ;
9
10 : RCVERR     GOBBLE 1 NAK-CT +!    NAK-CT @ MAXERR >
11              IF CAN XOUT CR ." Exceeded max. errors."
12                  ABEND QUIT
13              ENDIF
14              NAK XOUT    ." NAK sent" CR ;
15 -->

```

```

SCR # 31
0 ( SEND & RCV files using MODEM7 protocol          EHR 850420)
1
2 : ?TERMABOOT  0 ?TERMINAL IF DROP KEY 24 =
3              IF CR ." Cancelled by Sysop" -1 ENDIF ENDIF ;
4 : NOREND     CR ." EOT rcvd"    ACK XOUT XFER CLOSE-FILE 255 =
5              IF ." Can't close " XFER OUT-NAME CR ENDIF ;
6 : RCV-SECT   0 CRC ! 128 0 DO TIMED-IN DUP -1 = IF EXIT ENDIF
7              DUP XFER ?BUFFER-ADDR I + C!    CRC @ SWAP CRC-16 CRC !
8              LOOP 0 ;
9
10 : ?HDR-OK    OVER 255 XOR = ;          ( sec# _sec# --- sec# f)
11 : NEXT-SECT  RSEC ! 1STSEC @ IF 0 1STSEC !
12                  ELSE 1 XSEC +!    ENDIF 0 ;
13 : ?WRAP     DUP 0= RSEC @ 255 = AND ;   ( sec# --- sec# f)
14 -->
15

```

```

SCR # 32
0 ( SEND & RCV files using MODEM7 protocol          EHR 850420)
1
2 : RCV-HDR          ( --- f; sector # in XSEC )
3   TIMED-IN DUP -1 = IF EXIT ENDIF
4   TIMED-IN DUP -1 = IF SWAP DROP EXIT ENDIF
5   ?HDR-OK
6   IF DUP RSEC @ >          ( > previous sector#?)
7     IF NEXT-SECT          ( ok, inc sect. counters)
8     ELSE ?WRAP            ( sector wrap?)
9     IF NEXT-SECT          ( ok, inc sect. counters)
10    ELSE DROP 0 ENDIF     ( previous sector again)
11    ENDIF
12  ELSE DROP -1 ENDIF ;
13 -->
14
15

```

Edmund Ramm, P.O.Box 38, D-2358 Kaltenkirchen, W.Germany

```

SCR # 33
0 ( SEND & RCV files using MODEM7 protocol          EHR 850420)
1
2 : RCV-CRC      TIMED-IN DUP -1 = IF EXIT ENDIF      ( CRC MSB)
3               TIMED-IN DUP -1 = IF SWAP DROP EXIT ENDIF ( CRC LSB)
4               SWAP CRC @ SWAP CRC-16 SWAP CRC-16 DUP IF ( not 0)
5               DROP -1 ENDIF ;
6
7 : SAVE-SECT   XSEC @ XFER 33 + ! XFER WRITE-RAND 0= IF CR
8               ." Can't write sector." ABEND QUIT ENDIF
9               13 EMIT ." Received # " XSEC @ 1+ . ;
10
11 : DISP-CHR    BASE @ >R HEX 2 .R R> BASE !
12               ." H rcvd, not ACK" CR ;
13 -->
14
15

```

```

SCR # 34
0 ( SEND & RCV files using MODEM7 protocol          EHR 850420)
1
2 : WTCRC              ( --- SOH)
3     BEGIN
4     ." CRC request sent." CR 67 XOUT
5     ?TERMA BORT IF
6         ABEND QUIT
7     ENDIF
8     TIMED-IN DUP SOH =
9     IF
10        EXIT
11    ELSE
12        DROP
13    ENDIF
14    AGAIN ;
15 -->

```

```

SCR # 35
0 ( SEND & RCV files using MODEM7 protocol          EHR 850420)
1
2 : RCV    RCV-SETUP WTCRC 1 1STRCV ! 1 1STSEC !
3 BEGIN ?TERMA BORT IF ABEND QUIT ENDIF
4 1STRCV @ IF 0 1STRCV ! ELSE TIMED-IN ENDIF
5 DUP NUL = IF DROP RESTART ENDIF
6 DUP CAN = IF DROP CR ." Remote Cancel." ABEND QUIT ENDIF
7 DUP EOT = IF DROP NOREND QUIT ENDIF
8 DUP -1 = IF DROP ." T-" RCVERR RESTART ENDIF
9 DUP SOH <> IF DISP-CHR RCVERR RESTART ENDIF DROP
10 RCV-HDR IF ." H-" RCVERR RESTART ENDIF
11 RCV-SECT IF ." S-" RCVERR RESTART ENDIF
12 RCV-CRC IF ." C-" RCVERR RESTART ENDIF
13 SAVE-SECT ACK XOUT 0 NAK-CT !
14 AGAIN ;
15 -->

```

Edmund Ramm, P.O.Box 38, D-2358 Kaltenkirchen, W.Germany

```

SCR # 36
Ø ( SEND & RCV files using MODEM7 protocol          EHR 85Ø42Ø)
1 : WAIT-ACK  Ø ERRS !  Ø NAK-CT !          ( wait for ACK or CAN)
2   BEGIN
3     ?TERMA BORT IF  ABEND QUIT  ENDIF
4     ERRS @  MAXERR >  NAK-CT @  MAXERR >  OR
5     IF  CR ." Exeeded max. errs."  CAN XOUT
6     ABEND QUIT  ENDIF
7     TIMED-IN
8     DUP  -1 = IF DROP 1 ERRS +! CR ." T-" RESTART ENDIF
9     DUP CAN = IF DROP CR ." Remote Cancel" ABEND QUIT ENDIF
1Ø    DUP ACK = IF DROP Ø NAK-CT ! EXIT ENDIF
11    DISP-CHR 1 NAK-CT +! EXIT
12    AGAIN ;
13
14 : READ-SECT  SSEC @ XFER 33 + !  XFER READ-RAND Ø= ;
15 -->

```

```

SCR # 37
Ø ( SEND & RCV files using MODEM7 protocol          EHR 85Ø42Ø)
1
2 : SEND-HDR    SOH XOUT  SSEC @ 1+ DUP  255 AND  DUP XOUT
3              255 XOR  XOUT  13 EMIT ." Sending # "  U. ;
4
5 : SEND-SECT   Ø CRC !    128 Ø DO XFER ?BUFFER-ADDR I + C@ DUP
6              XOUT  CRC @ SWAP CRC-16 CRC !  LOOP ;
7
8 : SEND-CRC    CRC 1+ C@ XOUT  CRC C@ XOUT ;
9
1Ø : WTCRCRQ   BEGIN  ?TERMA BORT IF ABEND QUIT ENDIF
11              ." Awaiting CRC request" CR  TIMED-IN DUP
12              CAN = IF DROP ABEND QUIT ENDIF
13              67 =
14              UNTIL ;
15 -->

```

```

SCR # 38
Ø ( SEND & RCV files using MODEM7 protocol          EHR 85Ø42Ø)
1
2 : END-SEND    ( terminate xmsn & wait for ACK)
3  XFER CLOSE-FILE DROP
4  BEGIN  CR ." EOT sent"  EOT XOUT  WAIT-ACK
5        NAK-CT @ Ø=
6  UNTIL  CR ." ACK rcvd" ;
7
8 : SEND-SETUP  SETSIO I3ØØ  LONG-TIMEOUT
9  XFER FILENAME  CR  XFER OPEN-FILE
1Ø  255 = IF ." No File" CR  QUIT ENDIF
11  XFER OUT-NAME ."  open: "
12  XFER GET-SIZE  XFER 33 + @  U.  ." records" CR
13  Ø SSEC !  Ø NAK-CT ! ;
14 -->
15

```

Edmund Ramm, P.O.Box 38, D-2358 Kaltenkirchen, W.Germany


```

SCR # 39
0 ( SEND & RCV files using MODEM7 protocol          EHR 850420)
1
2 : SEND      SEND-SETUP  SHORT-TIMEOUT GOBBLE LONG-TIMEOUT
3            WTCRCRQ  SHORT-TIMEOUT
4 BEGIN      ?TERMA BORT IF ABEND QUIT ENDIF
5            READ-SECT  IF END-SEND QUIT ENDIF
6            SEND-HDR  SEND-SECT  SEND-CRC
7            WAIT-ACK  NAK-CT @ 0= IF 1 SSEC +! ENDIF
8 AGAIN ;
9
10 -->
11
12
13
14
15

```

```

SCR # 40
0 ( TERM & ECHO          EHR 850528)
1 : (ECHO)  DUP DUP XOUT 127 AND DCONOUT 13 =      ( c --- )
2 IF 10 DUP XOUT DCONOUT  ENDIF ;
3
4 : KBD->V24  DCONIN -DUP IF DUP 5 =              ( --- )
5 IF DROP CR ." Leaving terminal mode" CR  QUIT  ENDIF
6 ?ECHO @ IF (ECHO) ELSE DOUT ENDIF  ENDIF ;
7
8 : (TERM)  SETSIO  I300 CR                        ( --- )
9 BEGIN ?COM IF XIN ?ECHO @
10 IF (ECHO) ELSE 127 AND DCONOUT  ENDIF  ENDIF
11 ?TXRDY IF KBD->V24  ENDIF
12 AGAIN ;
13
14 : TERM 0 ?ECHO ! (TERM) ;           : ECHO 1 ?ECHO ! (TERM) ;
15 ;S

```

Edmund Ramm, P.O.Box 38, D-2358 Kaltenkirchen, W.Germany

Quellenhinweise:

- (1) Robert Taylor, "SEND and RCV: A Forth Implementation of the XMODEM Protocol", Doctor Dobb's Journal # 83, September 1983 p. 66-81.
- (2) Andrew S. Tanenbaum, "Computer Networks", Prentice-Hall.
- (3) Claus Wilhelm, "Datenübertragung", Militärverlag der DDR.

BOLDFIELD LIMITED COMPUTING

Sussex House, Hobson Street, Cambridge.

Tel: Ramsey (0487) 840740

Ref: PMD/LW3/30FG

Date: 30th April 1984

Dear F.I.G. Member,

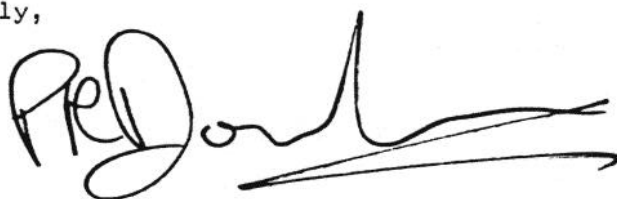
JUPITER ACE COMPUTERS

Although you have probably already heard the rumours, I am writing to you personally, because as a member of the FORTH INTEREST GROUP you will be especially pleased to hear the latest news regarding the Jupiter ACE.

The ACE is still the only cheap home computer dedicated to FORTH (and which incidentally also handles floating point arithmetic), and it is now only available by mail order from Boldfield - at remarkably low cost !

The enclosed brochure says it all, but remember to act quickly as stocks are obviously limited at these crazy prices.

Yours faithfully,



P.M. Downham
Director

PS. A whole new range of software is being launched this summer, and a host of exciting hardware add-ons become available soon. Although at very competitive prices, these items will not be sold as cheap as the current stock.

NETWORKING

ein Protokoll

From: JJAMES 26-MAR-1985 20:51
To: JJAMES
Subj: FIG Computer Network
to: FIG Business Meeting
from: John S. James (Delphi: JJAMES; msg: (408) 426-8988)
date: March 26, 1985
subject: FIG Computer Network

Status

(1) Local interest is growing. Members of two chapters have called me in the last month, about running their own computer locally.

(2) More FIG members have accounts on Delphi - I know of ten so far. There is also significant Forth activity on CompuServe, and on <Computer Languages> BBS in San Francisco.

Needed

(1) We should advise the chapters, and others who express interest, of options for getting involved at this time - both in national and local systems. What's needed is research and coordination, mostly by telephone, and preparing a writeup to send to local chapters. It must reflect what's available, and also what we want to do.

(2) Negotiations with Delphi and CompuServe should continue. We need to prepare a business plan for our use of the system.

Recommendations

I don't think we can recommend a single national system now. Delphi is best technically, but we don't know when it will have capacity for our SIG. Meanwhile, CompuServe offers a far larger audience, for anything we do which is directed outside the current FIG membership.

We shouldn't recommend a system to the general membership until negotiations are complete, our business plan is accepted, and we have started using the system so that there is something on it.

Instead, whoever expresses interest should be told what's happening on all the systems. Then he or she can make an individual choice.

Over the next few months, we should choose a single, official FIG national system. This choice will reflect the actual use which develops until then. There is no hurry, as long as we give those who do want to get started now the opportunity to do so.

NETWORKING

ein Protokoll

Goals

What do we want on the national and local systems? Here are some categories, mostly from the existing FIG Tree.

To make the national conference work best, we should be actively involved in what's on it. We should make sure that users who enter each category will quickly see useful, and recent, information. Less valuable or more specialized messages can also stay on the system, but they shouldn't be the ones seen first in their categories.

Here are my suggestions for categories, in the order in which they could be presented to the users:

(1) Introduction. A map or table of contents of what's where in the system, and how to find the highlights.

(2) What's happening. Calendar of events (either national or local). Forth activities online. Forth activities in other organizations. Forth in the press. Chapters may have their own sections here in the national system.

(3) Beginner's corner. Questions and answers at the level of <Starting Forth> and <Thinking Forth>.

(4) Systems. Discussion broken down by F83, MAC, MSDOS, and specific Forth products from vendors. Vendors can write intros for the discussion of their systems. The technical level should

(5) General discussion. Ideas, dialog. Education (may be its own separate area).

(6) Employment.

(7) <Forth Dimensions>. Pre-publication of articles for informal refereeing. Feedback on the printed issues. Support for columns such as Dear Doctor.

(8) FIG organization. Development of projects, and general comments and discussion.

(9) Users. Introduce yourself.

(10) Humor, and other non-Forth material.

How You Can Help

I will be very busy, and out of the area for several months. I can coordinate, but will need help. Most of the work is by phone or online. The immediate goal is to collect information and prepare a writeup to advise the chapters.

META FORTH PROCESSOR

Aus einem Interview mit
Alan Winfield

Meine Firma in Hull, England, heißt Metaforth Computer Systems Ltd. Wir produzieren und vermarkten eine sehr schnelle Forth-Maschine. Diese wurde gebaut im Anschluß an eine top-down Softwareanalyse verschiedener Forthsysteme. Das Ergebnis davon war etwas, daß wir "optimum forth engine" nennen. Diese wurde in Advanced Schottkey TTL implementiert. Wir haben kein Bit Slice Technik oder konventionelle Mikroprozessortechnik benutzt, weil wir diese für zu langsam halten. Unser Prozessor hat eine zugrunde liegende Zykluszeit von 50 ns. Das bedeutet, daß die mittlere Ausführungszeit im Bereich von 4 bis 5 Mio. Forthinstruktionen pro Sekunde liegt. Die Speicher sind in 3 unabhängige Bereiche aufgeteilt, den Datenstack, den Returnstack und das Mainmemory. Es ist ein 16-Bitprozessor. Die Datenwege und Stacks sind 16 Bit weit. Der Adressbus ist 24 Bit weit und die Maschine ist Byte-Adressed. Es ist also eine 16 Bit, Byte adressierte Maschine. Die beiden Stacks sind jeder 1K tief. Betrachtet man die Architektur, unterscheidet sich der Metaforth Prozessor vom Novix Prozessor dadurch, daß der Metaforth Prozessor mikroprogrammiert ist und der Novix Prozessor nicht. Das bedeutet, daß der Metaforth Prozessor tatsächlich Forth Instruktionen wie Opcodes im konventionellen Sinne abarbeitet. Wir haben die Mikro-Codierung vorgenommen, weil wir glauben, daß eine Maschine, welche eine Hochsprache ausführt, direkt komplexe Primitives ausführen können sollte. Wir haben diese Entwicklung begonnen als wir an der Universität ein Projekt über fortgeschrittene Computer-Architekturen durchführten. Und wir haben eine Forth-Architektur entwickelt, weil wir Forth besonders mochten. Forth hat uns begeistert und daher kam die Motivation diese schnelle Maschine zu entwickeln. Bisher wird sie nur in Großbritannien verkauft. An einem Vertrieb auf dem Kontinent wären wir interessiert.

(Vierte Dimension: Ein Artikel über die Metaforth-Maschine ist kürzlich in Bytes erschienen. Zum Vergleich hier die Daten für die Sieve Benchmark, 10 Iterationen: Metaforth 0,86 sec. gegen Ultraforth83 auf dem C64 mit 300 sec.! In der Welt sind zur Zeit mehrere Forthmaschinen in der Entwicklung. Zu nennen sind da insbesondere die Namen Chuck Moore in USA mit seinem Novix-Chip, Alan Winfield in GB mit dem Metaforth Prozessor und Bob Smith von Hartronics in USA.)

VD

MF16LP – SINGLE BOARD LANGUAGE PROCESSOR

Average performance: 4 Million High-Level Instructions per second

Preliminary Technical Specification

The *Metaforth MF16LP* is a single board 'language processor': a computer which does not use any conventional microprocessors but instead implements a completely new architecture optimised for the high-speed execution of high-level languages. The *MF16LP* does not have an assembly language - the high-level language 'Forth' is the lowest level at which the machine may be programmed. The *MF16LP* is by no means restricted to Forth, and will execute other high-level languages equally efficiently.

The *Metaforth* language processor will allow system designers with very demanding applications to achieve high-speed performance without having to resort to assembly language programming. Applications can be written entirely in high-level languages, giving significant benefits in reduced development times, portability, maintainability, ease of documentation etc., but without the performance degradation normally associated with the execution of high-level languages on conventional computers.

The Technology

With the exception of high-speed HMOS RAM devices, the *MF16LP* is implemented in Bipolar technology, with a number of custom Bipolar devices. This technology, together with its unique architecture and design, results in the high speed of the *MF16LP*.

Memory

The *MF16LP* is a 16-bit computer with a 24-bit address bus, giving an addressing range of 16 Megabytes. Memory is 16-bits wide, but with a dual word and byte addressing capability. The processor incorporates 2K words of high-speed stack and register memory.

Programming

The lowest level at which the *MF16LP* can be programmed is Forth, and to facilitate this the *Metasys* system software, supplied as standard, includes a full specification dual standard multi-tasking, Forth-79, Forth-83 system. In addition an integrated suite of development utilities is provided, including a disk filing system, a screen editor and an animator (allowing symbolic trace and single-step of programs under development). A full function 'C' compiler is also included with the software, as standard, and this is fully integrated into the Forth environment so that, if required, 'C' programs may include Forth segments and vice-versa.

© Metaforth Computer Systems Ltd 1985.

continued..

Performance

The majority of primitive Forth instructions (e.g. 16-bit addition, subtraction, stack manipulation; SWAP, DUP etc.) execute in under 200nS. Ultimately, however, the speed of the machine is determined by the speed of the external memory so that to achieve, for example, a maximum speed of 6.6 million Forth instructions per second requires external memory with a 60nS cycle time. In this situation the execution time for the primitive instructions ranges between 150nS and 550nS giving a performance of between 2 and 6.6 million Forth instructions per second. In real terms this means that the *MF16LP* will execute Forth programs 100 times faster than a conventional microprocessor implementation of Forth, and will approach a similar performance improvement for other advanced high-level languages including 'C', Pascal, and Modula.

Configuration Options

Three configurations of the *MF16LP* will be available: (1) the naked CPU card, (2) the CPU card together with RAM, ROM and I/O for stand alone evaluation purposes and (3) the CPU card with a standard VME bus interface.

(1) Naked CPU. The processor is mounted on a double Eurocard with a simple very high-speed bus interface. This allows optimum performance to be achieved with very simple memory and I/O designs and is thus ideal as the processor in a dedicated OEM application requiring cost effective high performance.

(2) Evaluation Unit. The *MF16LP* evaluation unit combines the language processor board with a memory, input-output and disk controller board - in a boxed unit together with floppy disk drives and power supply. The unit requires the addition of only a standard serial terminal.

(3) Full VME Development system. In the third configuration a piggyback memory and I/O card includes a multi-master VME bus interface so that the *MF16LP* may be used together with standard VME memory, input-output, disk interface or graphics controller cards. This configuration is thus ideal for substantial applications such as real-time control, vision processing, graphics, CAD/CAM or expert systems.

Processor Options

(1) Micro-programming. *Metaforth's 'Soft Architecture'* provides a mechanism for the rapid creation of new micro-coded primitive instructions; to accelerate the performance of applications software.

(2) High Speed Multiply. Provision is made for a high-speed hardware multiply device, giving a 100-fold speed improvement over the standard micro-code multiplication primitive.

Customer Support

Metaforth Computer systems are able to undertake Forth applications consultancy both in relation to this product and otherwise, and will provide assistance to the customer, if needed, in tailoring this product and its software to an application. We have special expertise in the following applications areas: languages and compiler design, communications, cryptography, vision processing, and expert systems.

Availability

The *MF16LP* will be available during the fourth quarter of 1985. *Metaforth* Computer Systems reserve the right to change this preliminary specification in order to further improve the product. VME, HMOS and C are registered trade marks and their use is acknowledged.

Tech.Spec.5 September 1985

The Metaforth *Soft Architecture*

Basic Concepts

Metaforth's 'Soft Architecture' provides a mechanism for the rapid creation of new micro-coded primitive instructions; to accelerate the performance of applications software.

In demanding real-time applications the conventional choices open to the designer are (1) the microprocessor solution or (2) the bit-slice solution. The non-microprogrammability and inflexibility of the microprocessor architecture forces the designer into compromising on performance, but the alternative bit-slice approach imposes a heavy cost penalty because of the considerable learning effort associated with micro-code programming.

Metaforth's solution to this dilemma is the provision of a processor architecture, in the *MF16LP* language processor, with a unique flexibility to be moulded to fit the demands of the application - through microcoding - but without the need for the designer to be intimately familiar with either the architecture or its micro-programming. Hence the '*soft*' architecture.

The Development Scenario

In practise the Soft Architecture concept will be exploited as follows:

1. The customer develops and tests the applications software, entirely in high-level language, using a *Metaforth* language processor with the standard firmware configuration and the base level set of hardware primitives. The software development process is hosted by the target language processor in conjunction with the customer's own applications specific hardware.
2. When the applications software is complete and debugged it is run, in the target environment, under the *Metaforth Profiler* in order to produce a detailed analysis of the activity of the application while running.
3. *Metaforth* engineers, in liaison with the customer, then use this analysis to identify and develop special additional hardware primitives. These new primitives may implement, in micro-code, frequently used functions originally defined entirely in high-level language, and/or special operations dealing with the customer's applications hardware. These are supplied to the customer as a new set of firmware ROMs, thus accelerating the run-time performance of the customer's application.

In essence this means that *you*, the customer, invent new primitive instructions using the *Profiler* as a tool for identifying these. *We* then design the micro-code for your new instructions, and test and install them. With *no* alteration of the applications software, its performance is enhanced.

Sal.5 June 1985 © Metaforth Computer Systems Ltd.

MF16LP Evaluation Unit

Technical Specification

The *MF16LP* evaluation unit combines the *MF16LP* language processor board with a memory, input-output and disk controller board - in a boxed unit together with floppy disk drives, cooling and power supply. The unit thus requires the addition of only a standard serial terminal. The principal function of the unit is to allow the language processor and its software environment to be fully evaluated, but the unit is useful also as a single or multi-user Forth development workstation.

The Processor

For a full description of the *MF16LP* language processor see '*MF16LP - Single Board Language Processor preliminary technical specification*', '*MF16LP: The Set of Primitives*' and '*MF16LP: Benchmark Results*'

The Memory Input-output and Disk Controller

This board incorporates,

1. 32k bytes upgradable to 128k bytes of high-speed (55nS) static main memory (RAM).
2. 512 bytes of high-speed boot ROM.
3. 16k bytes of standard EPROM containing the kernel system software. On reset this software is copied from the EPROM into RAM, where it is run at full speed.
4. 4 serial input-output channels organised as two dual UARTs. These allow individually software selectable baud-rates to 19.2k baud or synchronous serial communications to 1M bits/s. RS232 drivers are provided for each channel.
5. A disk controller (Western Digital 2797) to drive up to four standard 5.25" floppy disk drive units.

The Bus

The two cards are mounted on a short 3-slot backplane. Full details of the high-speed local bus are provided so that the customer may utilise the 3rd slot for applications hardware.

Disk Drives

Two half height 5.25" TEAC floppy disk drive units each provide 640kbytes of formatted storage.

Software

The *Metasys* system software supplied as standard includes a full-specification multiple standard multi-tasking and multi-user Forth system (Forth-79, Forth-83 etc.). In addition an integrated suite of development utilities is provided including a disk filing system, a screen editor, a profiler and debugging utilities. A full function 'C' compiler will be available, and this is fully integrated into the Forth environment so that, if required, 'C' programs may include Forth segments and vice-versa.

Availability

Orders are now being accepted for the Metaforth MF16LP evaluation unit, for delivery in October 1985. Please call for a firm quotation.

Eu1.5 © Metaforth Computer Systems Ltd, 1985.

Some Benchmarks and Timings for the

NOVIX Beta Board (NC4000P 6MHz)

These benchmarks are intended to give some comparisons with other processors running high level languages such as polyFORTH and 'C'. Bear in mind that the NC4000P figures relate to a 6MHz part - the production device will at 8MHz. Note: a * means these figures have been tested by COMSOL. Other figures published by Byte magazine or manufacturers.

1. 1,000,000 Empty Loops

68000 (8MHz) polyFORTH	18.0	secs	*
68000 (8MHz) Assembler	7.0	secs	
Intel 80286 (10MHz) 'C'	5.0	secs	
NC4000P (6MHz) novixFORTH DO...LOOP	2.4	secs	*
MF16LP (20MHz) FORTH	~ 1.0	sec	
NC4000P (6MHz) novixFORTH ..FOR...NEXT	170.0	msecs	*

2. Eratosthenes Sieve (x 10)

68000 (8MHz) polyFORTH	29.0	secs	*
Intel 80286/310 (10MHz) 'C'	6.6	secs	
68000 (8MHz) Assembler	4.9	secs	
VAX 780 'C'	1.4	secs	
CRAY-1 Fortran	1.1	secs	
MF16LP (20MHz) FORTH	1.09	secs	
NC4000P (6MHz) novixFORTH (cells)	0.85	secs	*
NC4000P (6MHz) novixFORTH (optimised)	0.45	secs	*
IBM 3033 PL/1	0.36	secs	
IBM 3081 PL/1	0.34	secs	

3. Fibonacci Series up to 24

8088 (IBM-PC) polyFORTH	19.0	secs	*
68000 (8MHz) polyFORTH	9.5	secs	
Sun 2/120 (68K) 'C'	1.7	secs	
Intel 80286/310 (10MHz) 'C'	1.2	secs	
NC4000P (6MHz) novixFORTH	0.19	secs	*

The remaining tests presented here give an idea of the speed of the NC4000P in some typical application areas.

4. POLLING I/O

This test program polls 1000 memory-mapped I/O devices and does a vectored execution of NULL if the I/O call is true.

Test 1 - no I/O lines set true - 1.15 msec to poll 1000 lines.

Test 2 - all I/O lines true requiring vectored execution of NULL
- 2.65 msec to poll 1000 lines.

5. Generation of Output Signals

We can use the NC4000 as a programmable waveform generator - for example:

SINE Wave Out Generation and Output

Using polyFORTH 14 bit fractional SIN arithmetic, the NC4000P can calculate the 14 bit sine value for every degree (i.e. 360 points for a full sine wave) to give an output frequency of 60Hz. A similar waveform generated using a 360 point look-up table runs at 1.66 KHz. The inner part of the code steps through in 7 cycles and is as follows:

```
....360 FOR I (SINE) @ Bport I! NEXT.....
```

RAMP Output

Using the FOR...NEXT construct the NC4000P can output a ramp via the Bport with one step every 3 clock cycles, as follows:

```
....FOR I Bport I! NEXT....
```

Multitasking

Task context switching time is given as 4.5 microseconds with an 8MHz clock.

Interrupts

The interrupt latency is between 2 and 4 clocks or, if streaming code is being executed by DO(...) then at the end of the stream.

October 1985

FORTH SYSTEME FÜR DIE INDUSTRIE

von Angelika Flesch

Produktivitätsgewinn in der Programmierung von Einplatinencomputer und Singlechip Rechnern: Die FORTH Methode

Die Oekonomie von Einplatinencomputern und Spezialprozessoren unterliegen anderen als den in PC's und Großrechnern gültigen Gesetzen. Speicher ist nicht im Überfluß erhältlich, jede benutzte Resource muß auch bezahlt werden.

Die Rechner die in der Zukunft weit mehr Prozessorleistung liefern werden, als alle Großcomputer und PC's zusammengenommen, werden die in Millionenstückzahl eingesetzten Prozessoren in intelligenten Arbeits- und Haushaltsgeräten sein: Angefangen bei intelligenten Backöfen, über Haus-Klimaanlagen Fahrzeugtechnik autonome Sensor/Effektorsysteme im Bereich der Umweltüberwachung, bis hin zu Raumfahrtcomputern. Die Speichergröße solcher Systeme wird aus vielen Gründen begrenzt sein. Einerseits, weil ein Single Chip Prozessor nur eine bestimmte Speichergröße zuläßt, andererseits, weil in Groß-serienstückzahlen ein Speicherchip mehr oder weniger einen gewichtigen Oekonomiefaktor darstellt, letztlich aber weil ein Bauteil, daß nicht eingebaut werden muß auch nicht defekt werden kann.

Die Verwendung von Hochsprachen zur Programmierung dieser Systeme stößt aber auf mehrere Hindernisse. Zuallerst ist der erzeugte Code wenig speicherplatz optimiert, andererseits erfordert die Verwendung einer Hochsprache einen Targetcompiler, der dann in vielen Fällen für den gerade verwendeten Prozessortyp nicht existiert, oder der zusammen mit dem benötigten Entwicklungssystem den Kostenrahmen des Projektes sprengt. Besonders kleine Labors bedienen sich daher der altbewährten Methode der Assemblerprogrammierung. Assemblercode ist schnell, kompakt und läßt sich in jedem Fall optimal auf die Gegebenheiten des Zielsystems anpassen. Die Nachteile der Assemblerprogrammierung sind aber ebenfalls bekannt: Programmiererproduktivität, die um eine Größenordnung unter der von Hochsprachen liegt, entsprechende Probleme in der Wartung und Pflege, überproportionaler Zeitaufwand bei steigender Komplexität, und als letztes fehlende Portabilität.

Eine Programmierung, die die Flexibilität und Portabilität von Hochsprachenprogrammierung mit der Kompaktheit und totalen Kontrolle der Assemblersprache kombiniert, und gleichzeitig ohne kostspielige Entwicklungssysteme auskommt, könnte den Engpass bei zur Zeit noch nicht ökonomisch bearbeitbaren Bereichen, wie z.B. in der Umwelttechnologie, überbrücken.

In der Tat existiert solch eine Methode schon: Das Programmsystem FORTH. Im Jahre 1967 von Charles Moore zunächst als Steuerungs-

FORTH SYSTEME FÜR DIE INDUSTRIE

von Angelika Flesch

system für Radioteleskope erfunden, hat diese Sprache zunehmend ihren Eingang in Einplatinencomputer und dedizierte Systeme, wie etwa den R65F11 Prozessor von Rockwell gefunden. Auch bei der NASA sowie in der Spaceshuttle werden einige Probleme mit Hilfe von FORTH Applikationen gelöst.

Das Arbeitsprinzip von FORTH:

Um solchen widersprüchlichen, und sich scheinbar gegenseitig ausschließenden Anforderungen genügen zu können, mußte der Erfinder von FORTH einige unkonventionelle Wege gehen. Das Grundprinzip von FORTH ist das eines Subroutinenprozessors: FORTH ist eine virtuelle Maschine die Subroutinen abarbeitet. Subroutinen werden aufgerufen indem einfach die Adresse der Routine abgelegt wird. Solch eine Adresse ist in der Regel 16 Bit breit, daher ist FORTH ähnlich kompakt wie Assemblercode. Da ein Prozessor, der die virtuelle FORTH Maschine emuliert, nicht ewig nur Adressen bearbeiten kann, gibt es auch einen Satz von Operationen, die in der Maschinensprache des verwendeten Prozessors kodiert sind, die also in der maximal möglichen Geschwindigkeit des Prozessors erstellt sind. Der Anwender kann aber auch eigene in Assembler geschriebene Befehlsfolgen zum FORTH-System hinzufügen. Dieser in Assembler kodierte Teil von FORTH ist sehr kompakt und umfaßt ca. 500 Byte bis 2 Kbyte. Solche Primitives lassen sich über die Fädelung von Subroutinenaufrufen beliebig zu komplexen Anweisungen kombinieren, damit wird der Grundsprachschatz um die Applikationsspezifischen Definitionen erweitert.

Dies ist ein unschätzbare Vorteil in der Entwicklungsphase einer Applikation auf Einplatinenrechnern. Damit können nämlich erste Designvorschläge für die zu verwendende Lösung direkt auf der Hardware des Zielsystems ausgetestet werden ohne gleich eine komplette Betriebsumgebung für das Gerät aufbauen zu müssen.

FORTH weicht sehr stark vom typischen Edit-Compile-Link-Go Zyklus der normalen Programmiersprachen ab. Der Editor ist Bestandteil des Interpreters und kann damit in Sekundenschnelle aufgerufen werden, der Sourcetext wird dabei in einem einfachen virtuellen Memory Schema zwischengebuffert, der Linkvorgang entfällt völlig und zum Aufrufen gelangen die neuen Definitionen einfach durch Eingabe des Befehlwortes in den Interpreter. Deshalb ist auch die normale turn-around-time in FORTH sehr viel geringer als in anderen Sprachen, d.h. Syntaxfehler sind sehr viel schneller ausgemerzt.

FORTH ist deshalb auch keine Programmiersprache im klassischen Sinn. FORTH wird am besten durch den Begriff "Programmierumgebung" beschrieben. Es benutzt ein eigenes Betriebssystem, das

FORTH SYSTEME FÜR DIE INDUSTRIE

von Angelika Flesch

an die eigenen Bedürfnisse angepaßt werden kann, und bietet Tools an die das Entwickeln und Austesten von Programmen sehr erleichtern. Diese Entwicklung und das Testen werden dabei immer interaktiv unterstützt. Die Fähigkeit, beim Austesten -so auf die schnelle- neue Worte oder Testsequenzen zu schreiben erlaubt das Aufbauen einer Programmierumgebung, die dem Problem, und der Denkweise des Programmierers über das Problem, entspricht.

Da die kompletten Utilities für die Fehlersuche, der Editor, ja sogar der FORTH Compiler größtenteils in FORTH selber geschrieben ist führt dies zu einer sehr großen Portabilität, Programmmodule sind auch nach dem Wechsel des Prozessortyps noch weiterverwendbar. Da diese Hilfsmittel ebenfalls in FORTH geschrieben sind fällt es auch einem FORTH Anwender leicht diese Programme an seine eigenen Bedürfnisse anzupassen und sogar Applikations spezifische Erweiterungen auch im Sprachkern einzuführen. In diesem liegt auch die Möglichkeit des Aufbaus einer Firmenspezifischen Programmierumgebung mit eigener Gestaltungsmöglichkeit. FORTH wächst dabei als Programmiersprache mit dem Programmierer mit, es macht die gleichen Erfahrungen: sein Wortschatz nimmt zu. Diese Anpassung an die Problemwelt, in der FORTH benutzt wird, ist auch für die drastische Reduktion der Programmentwicklungszeit verantwortlich.

FORTH bietet in einer einfachen integrierten Struktur die Fähigkeiten einer interaktiven Programmiersprache, eines Macro Assemblers und eines Operating systems mit Unterstützung durch Hilfsprogramme.

Auf welchen Grundzügen beruhen nun die wesentliche Vorteile von FORTH:

- 1.) **Erweiterbarkeit:** Standardmäßig wird FORTH mit ca. 30 - 40 maschinendefinierte und ca. 250 - 400 in FORTH definierte Befehls worte geliefert. Dieser Wortschatz wird vom Programmierer solange erweitert bis der Computer mit einem einzigen Befehl die gewünschte Aktion ausführen kann. Dies geschieht dadurch, daß die FORTH sprache der Applikation angepasst und kleine Bereiche der Applikation modular gelöst werden.
- 2.) **Stack orientiert:** FORTH arbeitet nicht mit Argumentlisten sondern stellt die Parameter über einen Stack zur Verfügung. Routinen oder Operationen kommunizieren dabei über einen Stack miteinander und benutzen dabei die gleiche Technik wie die Hewlett-Packard Taschenrechner, nämlich UPN (umgekehrte polnische Notation).

- 3.) **Strukturiert:** FORTH arbeitet ebenfalls strukturiert, d. h.

FORTH SYSTEME FÜR DIE INDUSTRIE

von Angelika Flesch

sie arbeiten mit Befehlsworten wie IF und ELSE, WHILE und REPEAT. Der Befehl GOTO wird nicht benötigt und ist deshalb auch nicht implementiert.

- 4.) **Wiedereinsprungsfähig:** Verschiedene Tasks oder auch User können den gleichen Code teilen, er ist voll reentry fähig.
- 5.) **Rekursiv:** FORTH routinen können sich selbst aufrufen und damit bestimmte Probleme einfacher lösen.
- 6.) **Schnell:** FORTH Programme sind in der Regel 5 bis 10 mal schneller wie andere Interpreter bzw. genauso schnell als andere optimierende Compiler. Typische Compilierungszeiten für eine 16 KByte Applikation beträgt ca. 10 Sekunden bis 3 Minuten (je nach Prozessor)
- 7.) **Kompakt:** FORTH Programme sind sehr kompakt, und damit auch mit 8 Bit Prozessoren voll ausnutzbar. Ein typischer interaktiver FORTH Kern mit Interpreter enthält ungefähr 8 KByte Code.
- 8.) **Multitasking fähig:** Verschiedene FORTHCompiler benutzen die Wiedereinsprungsfähigkeit um Multitasking Systeme aufzubauen. Dies ist zum Beispiel schon auf dem Z80 machbar. Auf 16 Bit Prozessoren wie 68000 oder 8086 ist dies natürlich effizienter machbar. Multitasking setzt einen mindestens 16 Bit breiten Stackpointer und Returnstackpointer voraus.
- 9.) **Assemblereinbindung:** FORTH erlaubt es sehr einfach Unterprogramme in Assembler einzubinden. Dafür wird ein im FORTH-compiler vorhandener umgekehrt polnisch arbeitender Assembler oder Hexcodeprogrammierung benutzt. Der Assembler benutzt strukturierte Befehlsworte wie IF, ELSE, WHILE ... REPEAT, zur Steuerung des Programmflusses. Durch FORTH-fähigkeiten können auch Macros und andere Hilfsmittel aufgebaut werden.
- 10.) **Interaktives Userinterface:** Diese Fähigkeit ermöglicht den schnellen Aufbau eines testbaren Benutzerschnittstelle, in der Testphase können dabei die Hilfsmittel des FORTH Interpreters genutzt werden ohne das dies programmiert werden muß.

Applikationen enthalten typischerweise nur höchstens 10 % zeitkritischen Code. Dieser Bruchteil läßt sich einfach mit dem FORTH Assembler programmieren. Für die restliche Applikation stehen dann die Fähigkeiten einer Highlevel Programmiersprache zur Verfügung. Diese problemlose Einbindung von Assembler-

FORTH SYSTEME FÜR DIE INDUSTRIE

von Angelika Flesch

routinen erlaubt es auch Interruptroutinen usw. in Highlevel FORTH zu erledigen.

Weitere Eigenschaften von FORTH:

FORTH als Programmiersprache eignet sich sehr gut für die Implementation auf einem Chip. Durch die in FORTH eingebaute Erweiterbarkeit des Wortschatzes, bei dem der Anwender in der Regel nicht unterscheiden muß ob der benutzte Befehl nun ein vorhandener oder ein zusammengesetzter Befehl ist, ist dies eine sehr gute Möglichkeit einen flexiblen für viele Einsatzzwecke spezialisierbaren Prozessor aufzubauen.

Diese Entwicklung begann mit dem RCA 1802/05 für den es als ROM Erweiterung einen kleinen FORTH Compiler/Interpreter gab. Später wurde dann von der Firma Rockwell die Single Chip Prozessoren R65F11 und R65F12 vorgestellt. Diese Bausteine enthielten auf dem Chip ein 2,5 K großes Maskenprogrammiertes ROM, Ein/Ausgabekanäle sowie einen seriellen Kanal. Mit Hilfe eines Development ROM's konnten dann in der Zielhardware direkt die Entwicklung betrieben werden. Diese Lösungen, obwohl sie z. B. Platz auf der Platine sparen, waren jedoch noch keine Prozessoren, die speziell darauf ausgelegt waren FORTH Code abarbeiten zu können, im Fall Rockwell sind es z. B. modifizierte 6502 Rechner.

Verschiedene andere Firmen haben jedoch schon seit einiger Zeit komplette FORTH Prozessoren in der Entwicklung, die dann FORTH Code sehr schnell abarbeiten können. Diese Bausteine realisieren dann die FORTH Maschine in Hardware. Die FORTH Maschine ist ein erweiterbarer RISC-Prozessor, ein Prinzip, das mit einem sehr einfachen Befehlssatz maximale Geschwindigkeit erzielt.

Zwei Firmen sind zur Zeit in der Lage FORTH Chips anzubieten und beide Chips ermöglichen in ihrer Maximalkonfiguration einen Programmdurchsatz von zehn Millionen FORTH Grundoperationen pro Sekunde (10 MIPS). Figur 1 zeigt den prinzipiellen Aufbau der NOVIX NC4000 Maschine und damit auch den grundsätzlichen Aufbau von FORTH.

Welche Hilfsmittel kann nun der normale Anwender für die Lösung seines Problems erhalten?

Auf den meisten verbreiteten Personalcomputern sind zur Zeit FORTH Compiler/Interpreter erhältlich. Der Leistungsumfang schwankt dabei jedoch erheblich. Um dann die erstellten Programme auf die Zielhardware zu bringen wird jedoch noch ein Metacompiler benötigt. Dieses Programm ist dabei etwas wie ein Compiler-Compiler, da das entstehende Zielprodukt ebenfalls wieder die Möglichkeit der Compilation haben kann.

FORTH SYSTEME FÜR DIE INDUSTRIE

von Angelika Flesch

Die Arbeitsweise mit einem Metacompiler ist dabei grob folgende:

- Festlegung der geplanten oder vorhandenen Memorymap, Aufteilung von RAM/ROM Bereichen, Festlegen der Peripherieadressen.
- Festlegung der Schnittstelle zum Testein-/ausgabegerät. Ueber diesen Kanal kann dann die interaktive Testphase laufen.
- Erzeugung eines Ein/Ausgabetreibers für den interaktiven Testkanal.
- Erzeugen eines interaktiven FORTH Kerns durch Einbau des Ein/Ausgabetreibers in den vorhandenen FORTH Compiler/Interpreter.
- Erster Test der Hardware, Erstellen von Testsequenzen für die weiteren Peripheriebausteine.
- Fähigkeiten des FORTH Compiler/Interpreters zur Datenübergabe über den Stack voll ausgenutzt werden.
- Einbau der getesteten Module in den Kern des Metasystems.

Diese Reihenfolge bei der Programmerstellung bewirkt, daß sie schrittweise ihr Problem lösen, dabei aber immer auf schon getestete Module zurückgreifen können. Das reduziert die Testzeit am Ende der Entwicklungszeit und hilft damit komplizierte Applikationen sicher aufbauen zu können.

FORTH SYSTEME FÜR DIE INDUSTRIE

von Angelika Flesch

Wie werden FORTH Sequenzen im Arbeitsspeicher dargestellt:

Nehmen wir eine spezielle Definition eines Wortes, daß einen Fotoapparat steuern soll. In FORTH benutzen wir dazu folgendes Statement:

```
: FOTO  OEFFNE VERSCHLUSS  ZEIT BELICHTEN  SCHLIESSE VERSCHLUSS ;
```

Nachfolgend eine Representation wie diese Definition im Speicher des Computers abgelegt ist.

! FOTO !	Name

! Link !	

! code pointer !	

! adr OEFFNE !	

! adr VERSCHLUSS !	

! adr ZEIT !	

! adr BELICHTEN !	

! adr SCHLIESSE !	

! adr VERSCHLUSS !	

! adr EXIT !	

Die Definition von z.B. BELICHTEN kann dabei auch wieder durch solch eine Liste erledigt sein, sie kann aber schon durch eine Primitivoperation wie das betätigen eines Ausgangs erledigt werden.

FORTH SYSTEME FÜR DIE INDUSTRIE

von Angelika Flesch

FORTH Benchmarks der Zeitschrift Computer Languages August 1985

Der komplette Artikel kann bei uns angefordert werden.

Benchmark results (sec)

Manufacturer and product	Loop	Subtraction test	Multiply test	Divide test	Move	Compare test	Sieve 1	Sieve 2	Sieve 3	Sieve 4	Sieve 5
IBM PC											
Forth Inc.											
polyFORTH II IBM	1.118	3.641	4.473	4.723	3.679	4.456	54.898	36.907	66.026	66.384	74.070
polyFORTH II MSD	1.063	3.470	4.250	4.463	3.471	4.235	52.502	35.214	62.855	63.195	70.510
Harvard Softworks HS/FORTH 2.05	1.204	3.284	4.134	4.594	3.636	4.131	47.818	33.841	53.530	56.934	61.522
Laboratory Microsystems PC/Forth 3.0	0.779	3.400	4.265	4.621	3.223	3.991	52.209	33.419	62.078	64.608	71.573
MicroMotion MasterForth 1.0	0.638	2.691	3.573	9.584	3.305	3.214	45.028	29.570	53.583	54.520	60.809
Mountain View Press MVP-FORTH 1	1.098	3.570	4.461	4.589	3.801	4.214	55.010	—	64.680	65.709	72.671
Next Generation Systems NGSForth 2.1.1	0.992	3.471	4.389	4.620	3.541	4.041	53.485	35.383	64.783	63.984	71.529
No Visible Support Software FB3 2.1.0	1.172	4.462	7.619	37.367	4.463	5.206	67.796	43.923	79.863	80.430	89.008
Shaw Laboratories TaskFORTH 8088 v. 1.00A	0.893	3.553	4.338	5.691	3.471	4.233	55.147	35.202	64.597	65.986	73.130
Sunset Technology ST-FORTH 1.3	0.850	3.470	4.303	5.130	3.471	4.049	54.754	34.498	64.808	64.511	71.953
Ubiquitous Systems u4th 1.0	—	—	—	—	—	—	—	—	—	—	—
Unified Software Systems UNIFORTH	0.779	3.426	4.281	4.462	3.371	4.061	54.145	34.128	63.807	65.141	71.826
Macintosh											
Creative Solutions MacForth 2.3	0.700	2.100	5.100	N/A	2.900	2.500	34.400	21.300	41.400	45.900	50.900
MicroMotion MasterForth 1.0	0.400	1.400	1.700	N/A	1.900	1.900	25.200	16.800	29.700	30.200	32.900

- N/A = Not applicable.
1. Running Sieve 2 crashes MVP-FORTH.
 2. Lacking a XENIX system for the IBM PC, u4th could not be tested or benchmarked.

Note: The benchmark routines used in this review are essentially identical with those used in a recent article in *BYTE* by Ernie Tello ("Software Review: polyFORTH and PC/FORTH," November 1984, pp. 303-314). The loop, subtract, multiply, divide, move, and compare tests originally appeared in *FORTH Dimensions* (issue III/1, p. 11). The various Eratosthenes Sieve programs are derivatives of the Sieve of Eratosthenes algorithm appearing in the September 1981 *BYTE*, which Tello selected for his review. Sieve 1 is

the basic Sieve of Eratosthenes algorithm. Sieve 2 is an improved Eratosthenes Sieve by Don Colburn. Sieve 3 is an Eratosthenes Sieve described as "optimized." Sieve 4 is a Sieve which makes use of a *CREATE DOES* character array. Sieve 5 is a Sieve described as using an "optimized array." Timings on the MS-DOS systems were performed by combining a count maintained by the 55-millisecond timer interrupt with low-order bits read directly from the IBM PC's 8253 chip, reset to mode 2. Although up to microsecond resolution is theoretically available using this technique, in practice resolution to about a millisecond was attained. Stand-alone polyFORTH resets the 8253 to interrupt every millisecond; in this case, polyFORTH's built-in timer utility—also estimated to have approximately millisecond resolution—was used.

Table 2.

Der Artikel wurde dem Industriekatalog der Firma Forth-Systeme Angelika Flesch entnommen. Ich danke der Firma Flesch für die freundliche Genehmigung. Zu Ihrer Information ist hier noch das Inhaltsverzeichnis des Kataloges wiedergegeben.

Sie bekommen den Katalog bei:

Forth-Systeme Angelika Flesch, Postfach 1226, 7820 Titisee-Neustadt
tel: 07651-1665

I N H A L T S V E R Z E I C H N I S

Einfuehrung.	1
LMI Systeme.	7
LMI Metacompiler	12
VAXFORTH32	26
NOVIX NC4000	27
Meta FORTH	36
Multi FORTH.	44
Digital Analog Simulation.	45
R6511 Karten	53
FORTH fuer Macintosh	61
FORTH fuer Atari 520 ST.	62
PC/FORTH fuer Apple IIe.	66
Apple MasterFORTH.	68
Apple PADS	69
Unix FORTH	71
Cyber Robot.	72
Olivetti M24	76
FORTH Buecher.	79
FIG.	94

Kleinanzeige

PDP 11/23 + von DEC

Seit einigen Wochen steht mir diese Maschine zur Verfügung, die ich auch privat nutzen kann. Sie können sich vorstellen, wie gerne ich auf ihr auch Forth betreiben würde. Als Amateur kommen für mich die im Handel befindlichen Versionen finanziell kaum in Frage, und da ich keine Maschinensprach-Erfahrung auf PDP habe, traue ich mir auch nicht zu, es selbst zu installieren.

Wer kann mir helfen?

Andreas Carl, Berlin, tel: 732586

Anzeige

=====

HHS

Z80A Karte mit FIG Forth !

32K ROM, 32K RAM, 2X Z80 PIO, 1X Z80 SIO, 2X A/D Wandler
wahlweise 8 oder 10 Bit weit, Uhr MM58174 National Semiconductor,
Watchdog-Timer.

FIG Forth mit voller I/O Unterstützung im EPROM.

Kostet?

Leerplatine + Forth EPROM + Handbuch 250.- DM
incl. MwSt. und Versandkosten.

Preise mit Bestückung sowie Mengenrabatte auf Anfrage.

Hirsch Hard- und Software
Kriegsstr.121, 75 Karlsruhe 1
Telefon 0271-854994

=====

Uwe Roth
Egelsbacherstr. 11
6000 Frankfurt 71

Gibt es bei der FIG eine auf den Commodore 3032 (evtl. auch 4032 oder 8032) angepasste FORTH-Version? Wenn ja - kann man diese von der FIG als Listing oder auf Kassette beziehen? Wenn nein - gibt es vielleicht eine allgemeine Version für 6502-Systeme? Was würde das kosten? Außerdem: Gibt es bereits FORTH-Versionen oder entsprechende Erweiterungen mit Fließkomma-Arithmetik?

DAS F83 MODELL

Aus einem Interview mit
Michael Perry

Das F83 ist eine Public Domain Implementation des Forth 83-Standards. Ich habe das System aus verschiedenen Gründen geschrieben. Als Henry dazu kam, hatte er einige andere Auffassungen darüber, aber wir haben uns weitgehend geeinigt. In der Hauptsache ging es um zwei Dinge. Einmal wollten wir den neuen Standard unterstützen, damit die Leute beginnen ihn zu verwenden. Deswegen haben wir ein Modell-System geschrieben, um aufzuzeigen, wie ein Standard-83-System aussehen kann. Und zum anderen wollten wir ein qualitativ gutes Forthsystem als freie Software veröffentlichen, damit es sich weit verbreiten kann und vielfältig angewendet wird.

Zudem wollten wir auch das Ansehen von Forth verbessern. Wir haben immer wieder Artikel gesehen wie den im Byte-Magazin, in denen es hieß: "Forth hat keinen Multitasker." Das ist absurd. Was sie meinten war: "FIG-Forth enthält keinen Multitasker." Und wir wollten nun zeigen, daß ein Forthsystem mit etlichem mehr an Fähigkeiten, mit File-Interface und solchen Dingen möglich ist und von jedermann benutzt werden kann. Gerade die Anfänger können so dem Forth einen ernstesten Versuch widmen und es mehr benutzen. Wir sind davon ausgegangen, daß ein kompletter Satz an Entwicklungshilfsmitteln dabei sein muß, denn sonst kann man ein System nicht richtig benutzen. Die alten FIG-Systeme hatten kaum einen vernünftigen Editor und keine anderen nennenswerten Tools. Deswegen haben wir einen Assembler, was sehr wichtig ist, dazugegeben und einen guten Editor, zumindest finden wir beide diesen einfach in der Benutzung, vielleicht eine sehr persönliche Auffassung. Ich weiß, daß es auch Leute gibt, die Visual-Editors gemacht haben, die weitergehen und auch in der Public-Domain zu bekommen sind, aber wir fanden unseren gut und einfach. Wir haben einen guten Debugger, der die Befehle im Singlestep durchläuft, dazugefügt, einen Multitasker, einen Metacompiler, den kompletten Quelltext, Kommentare und all diese Dinge, die ein System einfach benutzbar machen.

Mein Lieblingswerkzeug in dem gesamten System ist das VIEW Kommando, das mir sofort den Quelltext einer Definition zeigt. Dies finde ich äußerst wertvoll.

Bei der Entwicklung des Systems hofften wir, daß sehr viele Leute es benutzen werden, es weiter entwickeln, so daß es wächst und diese Entwicklungen Verbreitung finden. Wir sind ein wenig darüber enttäuscht, daß von den vielen Benutzern nur wenig Nachrichten zu uns zurückkommen. Wir würden gerne mehr von den Benutzern erfahren.

Eine ganze Menge zusätzlicher Dinge befinden sich in einer Sammlung in San Francisco, die von John Peters zusammengetragen wird. Er ordnet dort all diese Files auf Disketten und sortiert die Quelltexte. In dieser Sammlung sind verschiedene Visual-Editors, ein Modem-Programm, der Code für ein Bulletin-Board-System glaube ich, eine Cross-Referenz-Utility, die Ihnen sagen

DAS F83 MODELL

Aus einem Interview mit Michael Perry

kann, wie oft die Instruktionen DUP in einem bestimmten Wort benutzt wird, und eine ganze Reihe anderer kleiner Tools. Die meisten funktionieren ganz passabel. Probiert das einfach aus, verbessert einiges daran und gebt es in die Public-Domäne zurück. Wir möchten das System wachsen sehen. Ursprünglich haben wir das F83-System für die 8080, 8086 und 68000 CPU's entworfen. In der Zwischenzeit haben andere Leute das F83 in andere Umgebungen transportiert. Sehr gut geworden ist die Anpassung für die CPU 6809 unter Flex. Ron Rood hat dies gemacht.

Es ist äußerst kompatibel ausgefallen und wir sind sehr glücklich darüber.

Dr. Ting hat im letzten Jahr sein Buch über die innere Struktur des F83 geschrieben. Dieses ist ein ausgezeichnetes Referenzwerk. Wir haben daran gedacht, vielleicht auch ein Lehrbuch zu schreiben, das in die Benutzung des Systems einführt, welche Features es hat, wozu sie da sind und dererlei mehr. Außerdem denken wir zur Zeit daran ein 32-Bit-System zu schreiben. Aber das kann noch 2 Jahre dauern, sowas ist sehr viel Arbeit.

Um mit John Peters in Kontakt zu kommen, können Sie den FIG-Tree benutzen. Dies ist ein Bulletin-Board für die Forth Interest Group in Haywood, California. Die Nummer ist Areacode 415 538 3580, sodann zweimal CR, und sie werden mit 300 Baud eingeloggt. Es kostet nichts. Sie können dort Nachrichten für John hinterlassen. Er wird sich dann bei Ihnen melden.

Man kann uns erreichen über:
No Visible Support Software, Postbox 1344, 2000 Center Street,
Berkley California, 90704.

VD

: FORTH (S--)

Grafik for c't 80 ;

**Das Grafik - Paket
unter F83 - Kosten -
lose Information von**

ECKART W. F. SCHMIDT
AM BIRKENHOF 53
D 2000 NORDERSTEDT

PAD FORTH FÜR NS32000

von Thomas Asche

Erstmals wurde von der PAD-Datentechnik GmbH ein 32Bit FORTH für die Prozessorfamilie NS32000 mit dem Namen padFORTH 32 entwickelt. Mit der Programmierumgebung, die dem Programmierer mit dem padFORTH 32 angeboten wird, ist es nun möglich schnelle Problemlösungen mit einem leistungsfähigen Mikroprozessor und einer Programmiersprache, die die Vorteile eines Kompilers und eines Interpreters vereinigt, zu schaffen.

Ein Merkmal am padFORTH 32 ist die Darstellung einer normalen Zahl mit einer 32 Bit Länge und die Zahlen doppelter Länge umfassen sogar einen Wertebereich der durch 64 Bit dargestellt werden kann. Durch diese im padFORTH 32 implementierte Zahlendarstellung ist es möglich den gesamten Adressraum des Prozessors NS32000 als einfache Zahl darzustellen und beliebige mathematische und logische Operationen auf Adressen auszuführen.

Besondere Merkmale des padFORTH 32

- Standard 83

Bei der Entwicklung des Befehlssatzes des padFORTH 32 wurde die Empfehlung im FORTH-83 Standard des Forth Standard Teams vom August 1983 berücksichtigt. Dadurch ist eine weitgehende Kompatibilität zu bereits bestehenden FORTH-Systemen gegeben, und es wurde kein neuer Dialekt geschaffen.

PAD FORTH FÜR NS32000

von Thomas Asche

- Hohe Verarbeitungsgeschwindigkeit

Einer der herausragenden Merkmale des padFORTH 32 ist die große Verarbeitungsgeschwindigkeit des Compilers und Interpreters.

Dies wurde durch einige besondere Maßnahmen in der Implementation erreicht.

Zum Beispiel ist der oberste Wert auf dem Datenstack im Registersatz des Prozessors vorhanden. Zu dessen Manipulation wird meistens kein Datenfetchzyklus notwendig, und es wird oft nur ein Prozessorbefehl zur Datenmanipulation benötigt.

Für den Einsatz als Multiusersystem werden für jeden Benutzer eigene Datenbereiche zur Verfügung gestellt. Diese Datenbereiche werden durch den SB-Pointer des NS32000 angezeigt. Mit diesem Pointer im Registersatz des Prozessors können umfangreiche Adressierungen ausgeführt werden. Es sind also wenig Adressberechnungen notwendig.

- Relative Adressierung

Das padFORTH 32 ist vollständig relativ adressiert. Die Basisadresse, auf die die Forthadressen bezogen sind, steht im FP-Register ('FORTH-Pointer') des Prozessors. Dadurch ergibt sich kein Geschwindigkeitsverlust bei der Ausführung der Forthworte. Durch diese besondere Darstellung im Adressraum des

PAD FORTH FÜR NS32000

von Thomas Asche

Prozessors ist die Möglichkeit gegeben Programme in beliebige Speicherbereiche zu legen.

- Optimierte Bufferverwaltung

Zur Datenmanipulation auf dem Massenspeicher dienen im FORTH die Blockbuffer. Im padFORTH 32 werden diese Buffer so verwaltet, daß die Datenblöcke, die am häufigsten benutzt werden am längsten im Buffer bleiben. Dadurch wird ein optimaler Zugriff auf das Disketten- oder Harddisklaufwerk gewährleistet.

- Options

Neben einem Standardeditor (siehe "Programmieren in FORTH" von Leo Brodie) kann ein Floatingpoint-package, das den Floatingpointprozessor benötigt, und ein Realtime Multitasking Package, das den Interrupt Controller benötigt, integriert werden. Für die Kommunikation mit den Massenspeichern wie Hard-, Floppydisk oder Tape ist ein Disk Operating System (paDOS) in Vorbereitung.

C64 FORTH

Hes, Handic, Pfeifer, Hofacker

Für den Commodore C-64 gibt es neben anderen Programmiersprachen auch schon einige FORTH-Systeme. Vier davon habe ich erworben und diese getestet. Man möge mir bitte einräumen, daß meine Beurteilung etwas subjektiv ist, da ich meine FORTH-System voll und ganz für die Entwicklung von Steuerungen verwende. Deshalb lege ich wenig Wert auf die gebotenen Möglichkeiten bezüglich Grafik und Sound auf dem C-64.

Ich möchte mit diesem Artikel zwei Disketten-FORTH und zwei FORTH-ROM-Module beschreiben, wie folgt:

1. HES-FORTH ROM-Modul für den C-64:

ROM-Bereich: von 8000 hex bis BFFF hex, also 16 K-Byte

PREIS: ca. DM 200.--

Dokumentation: in Englischer Sprache (2000-Worte-Englisch), wenig Einführung, aber ausreichend und vollständig.

Beinhaltet: FORTH-Vokabular
FORTH-Assembler
FORTH-Screen-Editor
FORTH-Debug-Utility
FORTH-System-Vektoren und Worte
für Disk und Drucken
FORTH-Grafik für LowRes-Grafik
und Sprites
FORTH-Sound-System für den C-64 SID

- 1.1 Das FORTH-VOKABULAR entspricht im allgemeinen dem FIG-Standard mit einigen Abweichungen, es wird auch auf das FORTH-Buch von Brodie Bezug genommen, was die Einarbeitung, falls man Englisch kann, sehr erleichtert.
- 1.2 Der FORTH-ASSEMBLER ist der 6502-Assembler von Ragsdale
- 1.3 Der FORTH-SCREEN-EDITOR ist eine sehr überzeugende Leistung des HES-FORTH, da er die Möglichkeiten des C-64 voll ausnützt.

C64 FORTH

Hes, Handic, Pfeifer, Hofacker

Die FORTH-SCREENS werden im RAM unter dem Betriebssystem abgelegt, und umfassen somit nur 16 k-Byte, also nur 16 Screens pro File. Auf der Diskette werden die Source-Screens als REL-Files abgelegt. Es lassen sich auch FORTH-Maschinen-Module auf der Diskette abspeichern, die man dann als PGM-Files wiederfindet.

Ein Screen umfaßt 16 Zeilen zu je 64 Zeichen, da der C-64-Bildschirm aber nur 40 Zeichen breit ist, wird er in horizontaler Richtung gescrollt.

Man kann diesen Editor mit dem Standard-FORTH-Befehlsatz betreiben, genau so wie mit einer Untermenge von Wordstar-(TM)-Befehlen. Man kann aber auch in zwei Insert-Modi mit dem C-64 Bildschirm-Editor arbeiten, wobei die Funktionstasten F1 bis F8 sehr sinnvoll belegt sind.

Es seien nun die beiden Insert-Modi und die Funktionstastenbelegung kurz beschrieben. In den Insert-Mode 1 gelangt man durch erstmaliges Drücken der INST-Taste, ein zweitemaliges Drücken, und man ist im Insert-Mode 2, danach wiederum 1, u.s.w., mit RETURN verläßt man beide Insert-Modi.

- 1.31 Insert-Mode 1: Jedes von der Tastatur eingegebene Zeichen wird vor ein vorhandenes Zeichen platziert, und alle Zeichen in derselben Zeile um eine Position nach rechts weiter geschoben.
- 1.32 Insert-Mode2: jedes von der Tastatur eingegebene Zeichen wird über ein vorhandenes Zeichen platziert, alle Zeichen bleiben auf ihren Plätzen.
- 1.32 Funktionstastenbelegung: F1 Tab 4 Positionen rechts
F3 Tab 4 Positionen links
F5 entspricht Carriage-Return
F7 Find-Commando

F2 Editiere nächsten Screen
F4 Editiere vorhergehenden Screen
F6 Leerzeile einfügen
F8 Replatziert-Commando
- 1.4 Die FORTH-DEBUG-UTILITIES umfassen TRACE, SINGLE-STEP, DECOMPILE=
= SOURCE und DUMP.

C64 FORTH

Hes, Handic, Pfeifer, Hofacker

- 1.5 Die FORTH-SYSTEM-VEKTOREN und WORTE ermöglichen den Zugriff zur Diskette und die Ausgabe auf den Drucker mittels HI-Level FORTH-Worten.
- 1.6 Die FORTH-WORTE für GRAFIK und SOUND sind zwar leistungsfähig, aber von mir kaum erprobt.
- 1.7 Zusammenfassende Beurteilung:

Das HES-FORTH-Modul für den Commodore C-64 ist ein leistungsfähiges Komplet-FORTH, wenn man davon absieht, daß die Dokumentation nur in Englischer Sprache ist, nur 16 Screens je File angelegt werden können, und leider kein Directory von der Diskette zu haben ist, wenn man das HES-FORTH-Modul eingesteckt hat, die im Handbuch abgedruckte Routine arbeitet nicht.

PUSH



C64 FORTH

Hes, Handic, Pfeifer, Hofacker

2. HANDIC-FORTH ROM-Modul für den C-64:

ROM-Bereich: von 8000 hex bis BFFF hex, wovon jedoch nur 12 k-Byte benutzt sind

PREIS: ca. öS 2000.--

Dokumentation: englisch, und sehr dürftig, enthält nur Texte für die Fehlermeldungen (ab CR #4), das FORTH-GLOSSARY und die FORTH-79 Definitionen.

Es wird auf das PET-FORTH Manual verwiesen, das als Text-Buch und als User-Referenz charakterisiert wird.

Beinhaltet: FORTH-Vokabular
FORTH-Assembler
FORTH-Screen-Editor
FORTH-System-Worte für Disk und Drucker

2.1 Das FORTH-VOKABULAR entspricht im allgemeinen dem FIG-Standard mit einigen Abweichungen. Dieses FORTH sollte das PET-FORTH sein, nur das es auf dem C-64 lauffähig ist.

BOOT-UP ladet den ersten Screen und damit läßt sich das ganze System von der Diskette aus initialisieren. Sehr nett finde ich, daß im Anhang das Vokabular für den FORTH-79 Standard definiert ist, welchen ich zusammen mit den CASE-WORDS eben durch BOOT-UP laden lasse.

DO-LOOP Indizes stehen bis zur dritten Ebene zur Verfügung, welche sind: I, J, K

2.2 Der FORTH-ASSEMBLER ist der 6502-Assembler von Ragsdale.

2.3 Der FORTH-SCREEN-EDITOR arbeitet sowohl mit dem Standard-FORTH-Befehlssatz, als auch mit dem C-64 eigenen Bildschirm-Editor.

HANDIC-FORTH hat - im freien RAM ab HEX C000 - drei Screens als Puffer-Speicher und es stellt sich als ein 1000-Byte Screen dar mit 25 Zeilen zu je 40 Zeichen, es ist demnach dem C-64 voll angepaßt.

C64 FORTH

Hes, Handic, Pfeifer, Hofacker

Die Source-Screens werden auf der Diskette einer nach dem anderen abgespeichert, sie erscheinen jedoch nicht im Directory der Diskette. Auch wird die BAM (Block Availabilitz-Map) der Diskette nicht upgedated. Das heißt, für das CBM-Dos existieren diese Daten nicht, deshalb muß man diese Diskette sorgsamst von den anderen Disketten trennen. Andererseits kann man die Diskette fast vollständig mit Source-Screens füllen.

- 2.4 Die FORTH-WORTE für Disk-I/O sind einerseits die Standard-FORTH-Worte oder C-64 spezifische OPEN, CLOSE ua. FORTH-Worte.

Die Druckerausgabe wird nach Aufruf des EDITORS mittels PAPER eingeschaltet und mittels VIDEO ausgeschaltet.

- 2.5 Zusammenfassende Beurteilung:

Das HANDIC-FORTH-Modul für den Commodore C-64 ist ein leistungsfähiges Standard-Forth, wenn man von der äußerst spärlichen Dokumentation absieht. Vorteilhaft ist die BOOT-UP Funktion, womit man ein sofort arbeitendes FORTH-System nach seinem Geschmack erhält.

Negativ zu beurteilen ist die Tatsache, daß die Source-Screens nur durch persönliche Achtsamkeit des Anwenders geschützt sind, da im CBM-DOS nur durch Zusammenhängen der einzelnen Datenblöcke (Sektoren) mittels einer Directory-Eintragung die Files durch ungewolltes Überschreiben geschützt sind.

Ich möchte hiermit feststellen, daß ich mit diesem Modul am liebsten arbeite, weil ich einen guten FORTH Grundwortschatz habe, und mir die BOOT-UP Funktion sehr zustatten kommt.

C64 FORTH

Hes, Handic, Pfeifer, Hofacker

3. JÜRGEN PFEIFER Disk-FORTH für den C-64:

RAM-Bereich: ab 800 hex (Basic-Programm-Start)

PREIS: ca. DM 100.--

Dokumentation: deutsch/englisch, ausreichend

Beinhaltet: alles was man sich denken kann!

FORTH-Vokabular
FORTH-Assembler
FORTH-Screen-Editor
FORTH-Decompiler & Dump
FORTH-CBM-DOS-Link
FORTH-Grafik auch für Hi-Res-Grafik

und vieles vieles andere mehr,
und das in 121 Source-Screens!

3.1 Das FORTH-VOKABULAR entspricht dem FIG-Standard, und das ist das einzige Modul des gesamten FORTH, das in Object-Form vorliegt. Alles andere liegt in Source-Form in den 121 Screens vor.

3.2 Der FORTH-ASSEMBLER ist der 6502-Assembler von Ragsdale. Screens #20 bis #25, Demo auf SCR #26

3.3 Der FORTH-SCREEN-EDITOR bedient sich des C-64 Bildschirmformates 25 Zeilen zu je 40 Zeichen. Screens #12 bis #19.

Im Gegensatz zum HANDIC-FORTH sind hier die Screens (wahrscheinlich) durch Verkettung gegen ungewolltes Überschreiben geschützt. Dennoch rate ich von einem Validate oder Collect-Lauf der Diskette dringendst ab.

3.4 Der FORTH-DECOMPILER und FORTH-DUMP neben anderen interessanten FORTH-Wörtern sind unter den verschiedensten Gruppierungen auf den einzelnen Screens untergebracht, ich habe hier kaum Platz, dies alles aufzuzählen.

C64 FORTH

Hes, Handic, Pfeifer, Hofacker

- 3.5 Das FORTH CBM-DOS DISK-INTERFACE sollte nach der Beschreibung alle CBM-Disk-Drives bedienen können. Mit einer einzigen 1541 läßt sich dies jedoch kaum erproben.
- 3.6 Das FORTH GRAFIK-Module bedient auch die High-Resolution Grafik (320 x 200 Grafik-Punkte) auf den Screens #60 bis #63 mit Grafik-Demo auf Screens #65 bis #69.
- 3.7 Sonstige, äußerst nützliche FORTH-Module des Pfeifer-FORTH:

- Vektorisiertes I/O
- Doppelte Genauigkeit
- Fix-Point Trigonometrie
- Fix-Point Quadrat-Wurzel
- String-Verarbeitung

- 3.8 Zusammenfassende Beurteilung:

Das Jürgen Pfeifer Disketten FORTH für den C-64 ist ein leistungsfähiges, überkomplettes Standard-FORTH, eine wahre Fundgrube für den FORTH-Neuling und das vor allem deshalb, weil dieser damit in der Lage ist, FORTH - so wie es ist - im Source-Code zu analysieren. Kenner der "FORTH-DIMENSIONS" werden zwar viele der Module wieder erkennen, doch hat sich auch jener sein Geld redlich verdient der sich hinsetzt, und durch seiner Hände Werk anderen Erkenntnisse zu vermitteln, die diese sonst nicht erhielten. Die meisten FORTH-Anwender definieren ihren Code so kompakt als möglich, und meistens ohne Freigabe des Source-Codes. Ruhmreiche Ausnahmen bestätigen nur die Regel.

C64 FORTH

Hes, Handic, Pfeifer, Hofacker

4. W. HOFACKER Disk-Forth für den C-64:

RAM-Bereich: ab 800 hex (Basic-Programm-Start)

PREIS: Paket-Preis DM 149.--

beinhaltet: 2 Disketten, welche sind:

FORTH-System
SOURCE-Screens

und: 2 HOFACKER-Bücher, welche sind:

Band	137	FORTH-Einführung
Band	200	FORTH-Anwendungsbeispiele

Dokumentation: deutsch Hofacker-Bücher und die
beiliegende Kurzanleitung für den Editor und
für die Utilities
englisch Kurzanleitung für den Ragsdale-
Assembler
Wiedergabe des Artikels aus den
FORTH-Dimensions.

Beinhaltet: FORTH-Vokabular
FORTH-Assembler
FORTH-Screen-Editor
FORTH-Utility-Words
FORTH-WORDS für C-64 Grafik
und C-64 Sound

- 4.1 Das FORTH-VOKABULAR entspricht im Allgemeinen dem FIG-Standard und es ist das einzige Modul in Maschinenformat.
- 4.2 Der FORTH-ASSEMBLER ist der 6502-Assembler von Ragsdale und liegt als Source-Code vor (SCR# 40 ... SCR# 46)
- 4.3 Der FORTH-Screen-Editor ist der FIG-FORTH Line-Editor aus dem Installations-Manual, mit 16 Zeilen zu je 32 Zeichen realisiert; somit hat ein HOFACKER Source-Screen nur 1/2 kByte Umfang. Da dieser Editor ebenfalls in Source-Code vorliegt, dürfte er leicht an allgemeine FORTH-Verhältnisse anpaßbar sein. Man findet den Source-Code ab SCR# 7 vor.

C64 FORTH

Hes, Handic, Pfeifer, Hofacker

4.4 Die FORTH-Utilities liegen ab SCR# 21 mit folgenden Funktionen:

DUPLICATE	dupliziert Source-Screens von einer Diskette auf eine andere, mit händischem Disk-Swap
COPY (fr to)	kopiert SCR# fr in SCR# to
DUMP (adr n)	gibt n Speicherinhalte von adr ab auf aktivem List-Device aus.
PRON	schaltet alle Daten-Ausgaben auf den Drucker
PROF	schaltet alle Daten-Ausgaben auf den Monitor
OPEN (a n f d s)	öffnet Kanal f zum Gerät d mit Sekundäradresse s. Der Text von a mit der Länge n wird als Befehl übergeben, dazu dient " (--- a n) das wiederum mit " abgeschlossen wird, so kann man z.B. mit der Befehlssequenz " NEW0:NAME,BZ" 15 8 15 OPEN eine Diskette mit dem Namen "name" und der Prüfziffer "bz" neu formatieren.
CLOSE (n)	schließt den Datenkanal n.
INPUT (n)	die Eingabe erfolgt über den Datenkanal n.
OUTPUT (n)	die Ausgabe erfolgt über den Datenkanal n.
TERMINAL	die Eingabe erfolgt über die Tastatur, und die Ausgabe erfolgt über den Monitor.

4.5 Die FORTH-Worte für Sprite-Grafik befinden sich auf den Screens 49 und 50

4.6 Die FORTH-Worte für die Tongeneratoren befinden sich auf dem Screen 80, mit Demo-Sound auf Screen 81.

C64 FORTH

Hes, Handic, Pfeifer, Hofacker

4.7 Zusammenfassende Beurteilung:

Das Hofacker- bzw. Ekkehard Flögel-FORTH ist wohl das schwächste seiner Art, das auf dem Markte ist. Die 1/2 kByte-Screens und die einfallsslose Implementierung des Editors - der zum Glück im Source Code vorliegt - sind wiederum ein beredtes Zeugnis für die bekannte HOFACKER-Qualität. Für den Einsteiger ist es insofern interessant, da das meiste - zum Glück - in Source-Code vorliegt, doch empfehle ich dazu zum Jürgen-Pfeifer FORTH das ebenso fast alles in Source-Code vorlegt.



Einer unter Euch wird mich verraten.

Auswertungsbogen

Forth Systeme für Lernende

(aus FORTH Dimensions, Vol.VI, No.5, Seite 12)

Zum Lernen geeignet ist ein Forth System mit sieben Punkten oder mehr.
Bitte schicken sie eine Kopie ihrer Auswertung an: "Ask the Doctor",
Forth Dimensions, P.O.Box 8231, San Jose, California 95155.

Produkt:.....

Hersteller:.....

Händler:.....

Läuft auf:.....

Preis:.....

- _____ Objekt Größe (einen Punkt für je 5K bis zu 20K)
- _____ Optionen in Source Form (einen Punkt wenn diese mindestens 10 Blocks umfassen)
- _____ Editor (einen Punkt falls er mit dem in Starting Forth übereinstimmt; keinen Punkt für einen nach FIG-Art)
- _____ Editor Display (Scrolling giebt keinen Punkt; Screen gibt einen Punkt)
- _____ Massenspeicher (kein Punkt für Tonbänder; minus drei Punkte wenn die Diskettenunterstützung fehlt; einen Punkt für Harddisk Unterstützung)
- _____ Dialect (zwei Punkte für Forth-83; einen Punkt für Forth-79; einen Punkt für Polyforth; keinen Punkt für FIG-Forth)
- _____ Absolute Übereinstimmung zu einem der genannten Dialecte: einen Punkt dazu.
- _____ Documentation (einen Punkt für mehr als 50 Seiten)
- _____ Unterstützung (einen Punkt wenn der Hersteller/Händler telefonische Anfragen beantwortet)
- _____ Assembler (ja/nein; keine Punkte)
- _____ Floating Point (ja/nein; keine Punkte)
- _____ Zugriff auf Host File (ja/nein; keine Punkte)

Bitte nennen Sie andere mitgelieferte Erweiterungen:

BUG IM HES FORTH64 MODUL

von Andreas Carl

Im "Instruction Manual" Seite 70 findet sich ein offensichtlich fehlerhafter Algorithmus zum Auslesen der Directory der 1541. Anbei also eine korrekte Version!

Head (--): liest den Diskettentitel ein.
Dir (--): liest das Directory ein.

(Vierte Dimension: Weitere Kommentare erbeten.)
VD

```
SCR # 1
0      ( DIRECTORY )
1
2      DECIMAL
3 : ACPTRS 0 DO SYSTEM ACPTR DROP LOOP ;
4 : DCLOSE SYSTEM 15 AND 224 OR DR# @ LISTEN
5      SECOND UNLSN ;
6 : .FTYPE DUP 132 = IF ." REL FORTH FILE" ENDIF
7      DUP 130 = IF ." PRG" ENDIF
8      DUP 129 = IF ." SEQ" ENDIF ;
9
10 : DIRON  SYSTEM CMD I↑
11      DCHAN @ 1 + 240 OR DLSN ↑ $ ↑ STYPE 13 CIOUT UNLSN
12      ?DERR DCHAN @ 1+ DTLK 142 ACPTRS ;
13 : DIROFF SYSTEM UNTLK DCHAN @ 1+ DCLOSE ;
14
15      -->
```

```
SCR # 2
0      ( CONT. )
1 : DISKNAME SYSTEM 18 0 DO ACPTR DUP
2      128 < IF EMIT ELSE DROP ENDIF LOOP
3      ." ," 6 0 DO ACPTR 127 AND EMIT LOOP ;
4 : CONTENT SYSTEM BEGIN ACPTR DUP 199 - WHILE
5      DUP 0= IF 29 ACPTRS DROP
6      ELSE CR 2 ACPTRS SPACE
7      15 0 DO ACPTR 127 AND EMIT LOOP
8      10 ACPTRS ACPTR ACPTR
9      256 * + 3 .R SPACE .FTYPE
10     ENDIF 1+ DUP 8 = IF DROP 0
11     ELSE 2 ACPTRS ENDIF REPEAT DROP ;
12 : HEAD DIRON CR 4 SPACES DISKNAME DIROFF CR ;
13 : DIR  DIRON CR 4 SPACES DISKNAME
14      CR CR ." FILENAME SECTORS TYPE" CR 88 ACPTRS
15      CONTENT DIROFF ;
```

SINCLAIR QL Forth von Computer One

von Jens Storjohan

Der Sinclair QL ist ein Rechner, der einen Kompromiss darstellt zwischen einem Homecomputer und einem professionellen PC. QL ist die Abkürzung für Quantum Leap, Quantensprung, unter dem sich die Werbebranche einen besonders großartigen Sprung vorstellt. Die Physiker denken darüber anders.

Herzstück ist eine CPU 68000, bei der der Datenbus nur 8 Bit breit ist. Daneben gibt es noch einen Einchip Prozessor zur Bedienung der Tastatur und der anderen E/A-Kanaele. Der QL enthält in der Grundversion 128 kB RAM und zwei Microdrives genannte Laufwerke für spezielle schon für den Spectrum eingeführte jeweils etwa 100 kB fassende Microcassetten. Ein Videomodulator ist eingebaut, ebenso zwei serielle RS232c-Schnittstellen, eine Anschlußmöglichkeit für ein aus QL's gebildetes Netz und ein zwei Anschlüsse für Joysticks. Der Systembus ist herausgeführt, so dass man Speichererweiterungen oder Peripherie wie Floppy-Steuerungen anschließen kann. Mitgeliefert werden vier recht komfortable Programme auf Microcassetten und ein dicker Ordner mit Dokumentation. Der Preis liegt mit ca. 1500 DM eher in der Nähe der Homecomputer-Preise. Die deutsche Version wird inzwischen auch angeboten.

Das eingebaute BASIC hat fast alle Schwächen des althergebrachten abgelegt und wird zu Recht SuperBASIC genannt. Man kann nun getrost auf GOTO und GOSUB verzichten. Stattdessen kann man, ähnlich wie im Fortran, Funktionen und Subroutinen mit lokalen Variablen definieren. Die Programmablaufsteuerung ist ähnlich strukturiert wie in Fortran77 möglich. Das SuperBASIC enthält eine Reihe von kleineren Erweiterungen, die einem das Programmiererleben einfacher machen. So werden Variablen vom Typ Zeichenkette, wenn sie in einem arithmetischen Ausdruck stehen, automatisch gewandelt. Bei Vergleichsbefehlen für Zeichenkettenvariable erhält man z.B. die (vernünftige) Reihenfolge "a2" > "a10". Was man vermißt, ist die Möglichkeit Namen von Unterprogrammen als Parameter zu übergeben. Das Betriebssystem QDOS enthält recht fortgeschrittene Möglichkeiten, Ein/Ausgabekanäle zu definieren, insbesondere ein schönes Window-System, das zusammen mit den Graphik-Befehlen den Programmierkomfort auf eine Ebene bringt, von der ich als Benutzer eines IBM-Großrechners unter dem Betriebssystem MVS nicht zu träumen wagte.

Die vier mitgelieferten Programme sind umfangreiche und komfortable Produkte der englischen Firma Psion, nämlich das Textverarbeitungssystem Quill, das Graphik-Programm Easel, das Kalkulationsprogramm (vom Typ Sreadsheet) Abacus und das Datenbanksystem Archive. Wie man aus England hören kann, ist die verbesserte Version 2.0 dieser Programme erheblich schneller als die erste Version.

Der vorliegende Bericht wurde mit Hilfe von Quill verfasst.

SINCLAIR QL Forth von Computer One

von Jens Storjohan

Leider ist Quill nicht in der Lage, die deutschen Sonderzeichen ae, oe, und ss zu verarbeiten, obwohl der QL sie in BASIC-Modus durch Tastenkombinationen erzeugen und auch auf dem Schirm wiedergeben kann. Vielleicht weiß da jemand einen einfachen Trick oder in der neuen Auflage der Software kommt eine Ergänzung. Oder die deutsche Version schafft Abhilfe.

Leider ist die Dokumentation des QL voll von Druckfehlern und insbesondere bei der Beschreibung des Betriebssystems sehr unvollständig. Sie enthält allerdings recht viel einführendes Material für Anfänger. Die Dokumentation der vier mitgelieferten Programme ist ordentlich.

Als Bewertung läßt sich sagen: Kein voll professioneller Rechner, aber viel Hard- und Software fürs Geld. Leider entspricht die Genauigkeit der Gleitkommazahlen nur knapp sieben Dezimalstellen. Die Geschwindigkeit der Microdrives ist zwar erheblich besser als etwa die von Audiocassetten, erreicht aber nicht die einer Floppy. Die Rechengeschwindigkeit des Basic ist etwa vergleichbar mit der eines schnellen Z80-Systems. Wenn man in Assembler oder Forth programmiert, erreicht man Ausführungsgeschwindigkeiten, die etwas höher liegen als die des IBM-PC.

Inzwischen ist gerade in seinem Heimatland die Produktion von zusätzlicher Hard- und Software richtig angelaufen. Es gibt schon Floppysysteme, verschiedene Spielprogramme und nicht zuletzt eine erstaunliche Vielfalt von Sprachen wie BCPL (genauso eine britische Spezialität wie lauwarmes Bier), Pascal, Assembler, Fortran und andere.

Für den Forth-Programmierer ist interessant, daß es für knapp 200 DM ein Forth von Computer One zu kaufen gibt. Es enthält einen Full-screen Editor mit Wordstar-ähnlichen Befehlen, eine Verbindung zu Dateien im Stile von LMI, Graphik, Gleitkomma, einen Assembler, ein einfaches Multitasking und ausserdem die Möglichkeit, sich sein Forth-System neu zu definieren und in übersetzter Form auf der Kassette abzulegen. Es spricht Windows, Graphik, die Microdrives u.s.w. mit ähnlichen Konventionen an wie das SuperBASIC.

Die Dokumentation für das Forth von Computer One ist mit siebzig Seiten äusserst knapp und läßt einen häufig sein Heil im Decompiler, der zusammen mit einigen Utilities mitgeliefert, suchen.

Auch hier läßt sich sagen: Viel Forth fürs Geld, aber wohl doch nicht für professionelle Anwendungen gedacht.

Ich selbst bereue es jedenfalls nicht, daß ich mir für rund 300DM ein Computersystem einschließlich Drucker, Monitor und Forth gekauft habe, das mit seinem reichen Satz von Graphik- und Gleitkomma-Befehlen recht gut für wissenschaftlich technische Anwendungen, aber nicht nur dafür, geeignet ist. VD

FORTH URLADEN

von Frank Kaiser

Ich bin noch nicht im Besitz eines Forth-Systems. Auch in Südergellersen bot sich mir keine Lösung dieses Problems an; auch ein für den VC-64 entwickeltes System läuft nicht auf meinem Rechner (VC-20), da ein Forth-Compiler im Betriebssystem schon vorhandene Maschinenprogramme nach Möglichkeit mitbenutzt, was insofern rationell ist, als sich das vorhandene Betriebssystem in ROM befindet.

(Ich konnte in der Zwischenzeit zwei Forth-Systeme für den VC-64 kopieren, die aber nicht auf meinem Rechner laufen. Falls jemand Interesse daran hat, dann soll er sich bei mir melden.)

Ich habe mich deswegen jetzt entschlossen selber ein Forth-System auf meinem Rechner zu implementieren.

Zu diesem zunächst komplex erscheinenden Thema fand ich in der Bibliothek des Fachbereiches Informatik der Universität ein Buch (englisch) mit dem Titel: "Threaded Interpretive Languages". Es handelt sich um eine Beschreibung des Forth-Konzeptes auf der Systemebene und liest sich fast wie eine Bauanleitung für ein Forth-System, was auch für Neulinge (wie mich) verständlich geschrieben ist.

Ich bin zu der Überzeugung gekommen, daß es für ein optimales Arbeiten mit den Forth-Systemen des Wissens um den grundlegenden Aufbau eines solchen Systems bedarf. Deswegen schlage ich vor, in der geplanten Zeitschrift eine Serie über diesen Themenkomplex, Aufbau und Selbstbau eines Forthsystems, erscheinen zu lassen.

(Vierte Dimension: Forth wurde schon oft urladen. Es ist eine wahrhaft grundlegende Technik. Ich bin gespannt wieviele verschiedenen Methoden dazu bei mir eingehen werden. Bitte schreibt mir wie Ihr dieses Problem gelöst habt. Es wäre nett wenn die Urlader als Code mit dabei wären.)

VD.

LISTING IM F83

von Michael Kalus

Vor kurzem habe ich das FORTH 83 MODEL auf einem Apple II kompatiblen mit 56K CP/M 2.2 untergebracht. Als Anfänger im F83 hatte ich zunächst Probleme ein LISTING der Source Files auf den Drucker zu bringen. Rückblickend war es eigentlich einfach. Das ist eine der netten Seiten an FORTH, man kann sich selbst zurechtfinden mit etwas probieren und blättern im Source Code. FORTH finde ich sehr transparent. Das gilt besonders für das F83 von Henry Laxen und Michael Perry aus Californien/USA, weil hierbei die Quellen des ganzen Systems mitgeliefert werden. Mit den Worten SEE und VIEW kann man sich das F83 gut erschließen, und die Kommentare in den SHADOW Screens sind gut. So bin ich recht bald darauf gestoßen, daß man ein File komplett drucken lassen kann nachdem es geöffnet worden ist. Ein LISTING des 185 KByte langen File KERNAL80.BLK das den Quelltext des Forth Kerns enthält, erzeugt man so:

```
OPEN KERNAL80.BLK      LISTING <ret>
```

Und schon legt der Drucker los ... falls er angepaßt ist! Denn er muß dazu 132 Zeichen pro Zeile können. Und außerdem ist unser Papier hier in good old germany leider 72 Zeilen lang statt der 66 Zeilen kurzen Blätter in Californien. Also muß LISTING erst mal zurechtgebogen werden. Man betrachtet also mit VIEW LISTING den Source Code und hangelt sich von da weiter. Früher oder später findet man:

```
66 CONSTANT L/PAGE und
DEFERE INIT-PR ' NOOP IS INIT-PR
```

Da sind die Übeltäter. L/PAGE stellt die Papierlänge in Zeilen dar. Hier müssen wir also ran und die Constante im compilierten F83 ändern. Hirnchirurgie!

```
: PATCH-L/PAGE (n--) A'Ü L/PAGE >BODY C! : 72 Patch-L/Page
```

Diese Operation setzt die neue Seitenlänge schmerzlos ein. Man überzeuge sich davon:

```
SEE L/PAGE <ret>
72 CONSTANT L/PAGE ok
```

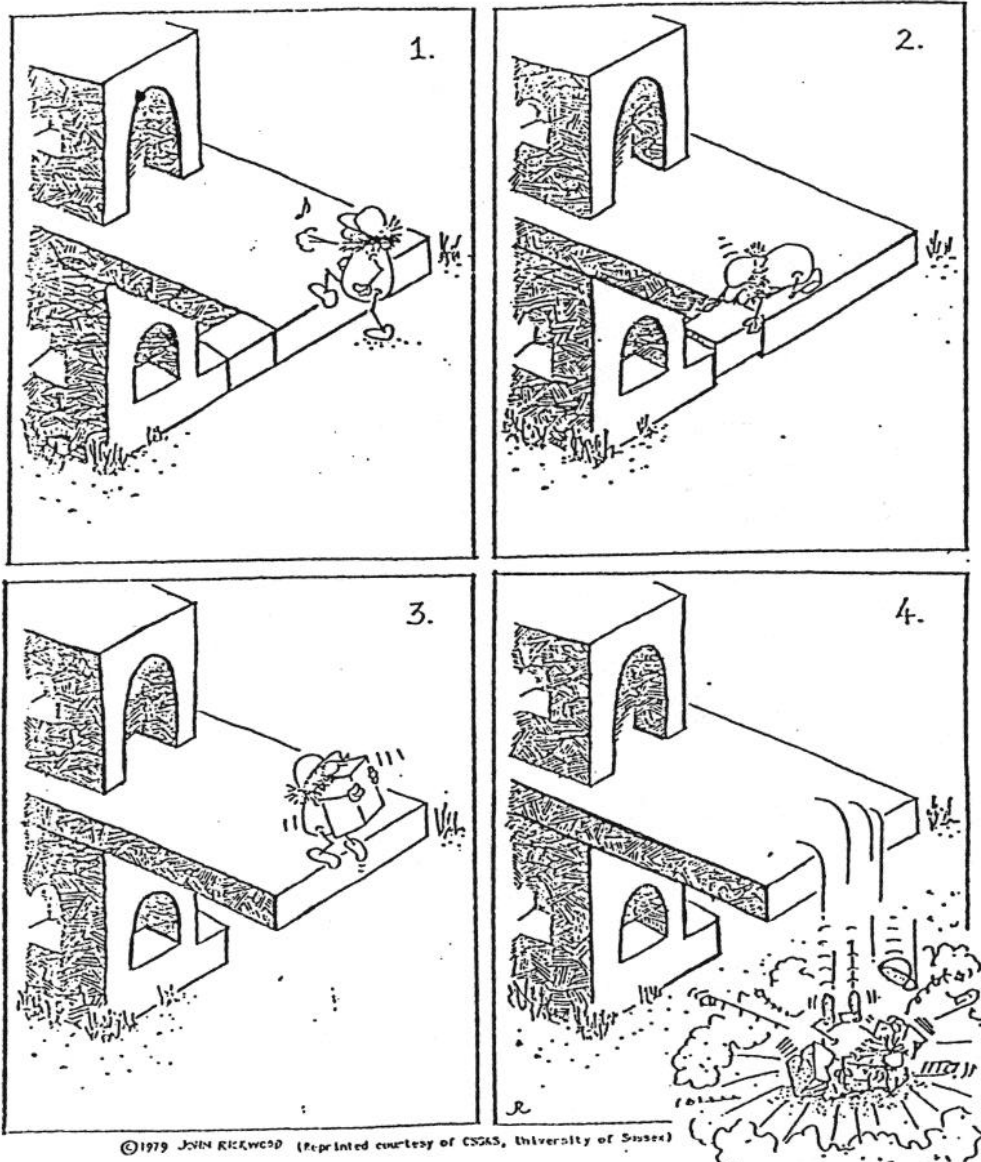
Nun noch den Drucker einstellen. Nicht nur hier waren die Autoren des F83 weitsichtig und haben die Drucker-Initialisierung vektorisiert. Ihren eigenen Epson-Drucker findet man noch im Source Code wieder, er wird mit EPSON IS INIT-PR in den Init-Vektor eingesetzt. Nun denn, flugs den eigenen Drucker statt dessen dort hingbracht:

LISTING IM F83

von Michael Kalus

: RITEMAN 15 (Compressed Characters) EMIT :
' RITEMAN IS INIT-PR

Von nun an wird mit 132 Zeichen pro Zeile und mit 72 Zeilen pro Seite das LISTING gedruckt, so wie es sein muß. Wie man sieht war es nicht schwer.



FORTH INTEREST GROUP MAIL ORDER FORM

P.O. Box 8231 San Jose, CA 95155 (408) 277-0668

MEMBERSHIP IN THE FORTH INTEREST GROUP

107 - MEMBERSHIP in the FORTH INTEREST GROUP & Volume 7 of FORTH DIMENSIONS. No sales tax, handling fee or discount on membership. See the back page of this order form.

The Forth Interest Group is a worldwide non-profit member-supported organization with over 5,000 members and 80 chapters. FIG membership includes a subscription to the bi-monthly publication, FORTH Dimensions. FIG also offers its members publication discounts, group health and life insurance, an on-line data base, a job registry, a large selection of Forth literature, and many other services. Cost is \$20.00 per year for USA, Canada & Mexico; all other countries may select surface (\$27.00) or air (\$33.00) delivery.

The annual membership dues are based on the membership year, which runs from May 1 to April 30.

When you join, you will receive issues that have already been circulated for the current volume of Forth Dimensions and subsequent issues will be mailed to you as they are published.

You will also receive a membership card and number which entitles you to a 10% discount on publications from FIG. Your member number will be required to receive the discount, so keep it handy.

HOW TO USE THIS FORM

1. Each item you wish to order lists three different Price categories:

Column 1 - USA, Canada, Mexico
Column 2 - Foreign Surface Mail
Column 3 - Foreign Air Mail

2. Select the item and note your price in the space provided.

3. After completing your selections enter your order on the fourth page of this form.

4. Detach the form and return it with your payment to **The Forth Interest Group**.

FORTH DIMENSIONS BACK VOLUMES

The six issues of the volume year (May - April)

101 - Volume 1 FORTH Dimensions (1979/80) \$15/16/18 _____
102 - Volume 2 FORTH Dimensions (1980/81) \$15/16/18 _____
103 - Volume 3 FORTH Dimensions (1981/82) \$15/16/18 _____
104 - Volume 4 FORTH Dimensions (1982/83) \$15/16/18 _____
105 - Volume 5 FORTH Dimensions (1983/84) \$15/16/18 _____
106 - Volume 6 FORTH Dimensions (1984/85) \$15/16/18 _____

ASSEMBLY LANGUAGE SOURCE CODE LISTINGS

Assembly Language Source Listings of fig-Forth for specific CPUs and machines with compiler security and variable length names.

513 - 1802/MARCH 81 \$15/16/18 _____
514 - 6502/SEPT 80 \$15/16/18 _____
515 - 6800/MAY 79 \$15/16/18 _____
516 - 6809/JUNE 80 \$15/16/18 _____
517 - 8080/SEPT 79 \$15/16/18 _____
518 - 8086/88/MARCH 81 \$15/16/18 _____
519 - 9900/MARCH 81 \$15/16/18 _____
520 - ALPHA MICRO/SEPT 80..... \$15/16/18 _____
521 - APPLE II/AUG 81..... \$15/16/18 _____
522 - ECLIPSE/OCT 82 \$15/16/18 _____
523 - IBM-PC/MARCH 84..... \$15/16/18 _____

- 524 - NOVA/MAY 81 \$15/16/18 _____
- 525 - PACE/MAY 79 \$15/16/18 _____
- 526 - PDP-11/JAN 80 \$15/16/18 _____
- 527 - VAX/OCT 82 \$15/16/18 _____
- 528 - Z80/SEPT 82 \$15/16/18 _____

BOOKS ABOUT FORTH

- 200 - ALL ABOUT FORTH \$25/26/35 _____
Glen B. Haydon
An annotated glossary for MVP Forth; a 79-Standard Forth.
- 205 - BEGINNING FORTH \$17/18/21 _____
Paul Chirlian
Introductory text for 79-Standard.
- 215 - COMPLETE FORTH \$16/17/20 _____
Alan Winfield
A comprehensive introduction including problems with answers. (Forth 79)
- 220 - FORTH ENCYCLOPEDIA \$25/26/35 _____
Mitch Derick & Linda Baker
A detailed look at each FIG-Forth instruction.
- 225 - FORTH FUNDAMENTALS, V. 1 \$16/17/20 _____
Kevin McCabe
A textbook approach to 79 Standard Forth.
- 230 - FORTH FUNDAMENTALS, V. 2 \$13/14/16 _____
Kevin McCabe
A glossary.
- 233 - FORTH TOOLS \$19/21/23 _____
Gary Feierbach & Paul Thomas
The standard tools required to create and debug Forth-based applications.
- 237 - LEARNING FORTH \$17/18/21 _____
Margaret A. Armstrong
Interactive text, introduction to the basic concepts of Forth. Includes section on how to teach children Forth.
- 240 - MASTERING FORTH \$18/19/22 _____
Anita Anderson & Martin Tracy (MicroMotion)
A step-by-step tutorial including each of the commands of the Forth-83 International Standard; with utilities, extensions and numerous examples.
- 245 - STARTING FORTH (soft cover) \$20/21/22 _____
Leo Brodie (FORTH, Inc.)
A lively and highly readable introduction with exercises.
- 246 - STARTING FORTH (hard cover) \$24/25/29 _____
Leo Brodie (FORTH, Inc.)
- 255 - THINKING FORTH (soft cover) \$16/17/20 _____
Leo Brodie
The sequel to "Starting Forth". An intermediate text on style and form.
- 265 - THREADED INTERPRETIVE LANGUAGES \$23/25/28 _____
R.G. Loeliger
Step-by-step development of a non-standard Z-80 Forth.
- 270 - UNDERSTANDING FORTH \$3.50/5/6 _____
Joseph Reymann
A brief introduction to Forth and overview of its structure.

FORML CONFERENCE PROCEEDINGS

FORML PROCEEDINGS - FORML (the Forth Modification Laboratory) is an informal forum for sharing and discussing new or unproven proposals intended to benefit Forth. Proceedings are a compilation of papers and abstracts presented at the annual conference. FORML is part of the Forth Interest Group

- 310 - FORML PROCEEDINGS 1980 \$25/28/35 _____
Technical papers on the Forth language and extensions.
- 311 - FORML PROCEEDINGS 1981 (2V) \$40/43/45 _____
Nucleus layer, interactive layer, extensible layer, metacompilation, system development, file systems, other languages, other operating systems, applications and abstracts without papers.
- 312 - FORML PROCEEDINGS 1982 \$25/28/35 _____
Forth machine topics, implementation topics, vectored execution, system development, file systems and languages, applications.
- 313 - FORML PROCEEDINGS 1983 \$25/28/35 _____
Forth in hardware, Forth implementations, future strategy, programming techniques, arithmetic & floating point, file systems, coding conventions, functional programming, applications.
- 314 - FORML PROCEEDINGS 1984 \$25/28/35 _____
Expert systems in Forth, using Forth, philosophy, implementing Forth systems, new directions for Forth, interfacing Forth to operating systems, Forth systems techniques, adding local variables to Forth.

ROCHESTER PROCEEDINGS

The Institute for Applied Forth Research, Inc. is a non-profit organization which supports and promotes the application of Forth. It sponsors the annual Rochester Forth Conference.

- 321 - ROCHESTER 1981 (Standards Conference) \$25/28/35 _____
79-Standard, implementing Forth, data structures, vocabularies, applications and working group reports.
- 322 - ROCHESTER 1982
(Data bases & Process Control) \$25/28/35 _____
Machine independence, project management, data structures, mathematics and working group reports.
- 323 - ROCHESTER 1983 (Forth Applications) . \$25/28/35 _____
Forth in robotics, graphics, high-speed data acquisition, real-time problems, file management, Forth-like languages, new techniques for implementing Forth and working group reports.
- 324 - ROCHESTER 1984 (Forth Applications) . \$25/28/35 _____
Forth in image analysis, operating systems, Forth chips, functional programming, real-time applications, cross-compilation, multi-tasking, new techniques and working group reports.

THE JOURNAL OF FORTH APPLICATION & RESEARCH

A refereed technical journal published by the Institute for Applied Forth Research, Inc.

- 401 - JOURNAL OF FORTH RESEARCH V.1 #1 \$15/16/18 _____
Robotics.
- 402 - JOURNAL OF FORTH RESEARCH V.1 #2 \$15/16/18 _____
Data Structures.
- 403 - JOURNAL OF FORTH RESEARCH V.2 #1 \$15/16/18 _____
Forth Machines.
- 404 - JOURNAL OF FORTH RESEARCH V.2 #2 \$15/16/18 _____
Real-Time Systems.
- 405 - JOURNAL OF FORTH RESEARCH V.2 #3 \$15/16/18 _____
Enhancing Forth.
- 406 - JOURNAL OF FORTH RESEARCH V.2 #4 \$15/16/18 _____
Extended Addressing.

REPRINTS

- 420 - BYTE REPRINTS \$5/6/7 _____
Eleven Forth articles and letters to the editor that have appeared in *Byte* magazine.
- 421 - POPULAR COMPUTING 9/83 \$5/6/7 _____
Special issue on various computer languages, with an in-depth article on Forth's history and evolution.

DR. DOBB'S

This magazine produces an annual special Forth issue which includes source-code listings for various Forth applications.

- 422 - DR. DOBB'S 9/82 \$5/6/7 _____
- 423 - DR. DOBB'S 9/83 \$5/6/7 _____
- 424 - DR. DOBB'S 9/84 \$5/6/7 _____

HISTORICAL DOCUMENTS

- 501 - KITT PEAK PRIMER \$25/27/35 _____
One of the first institutional books on Forth. Of historical interest.
- 502 - FIG-FORTH INSTALLATION MANUAL .. \$15/16/18 _____
Glossary model editor - We recommend you purchase this manual when purchasing the source-code listings.

REFERENCE

- 305 - FORTH 83 STANDARD \$15/16/18 _____
The authoritative description of 83-Standard Forth. For reference, not instruction.
- 300 - FORTH 79 STANDARD \$15/16/18 _____
The authoritative description of 79-Standard Forth. Of historical interest.
- 316 - BIBLIOGRAPHY OF FORTH REFERENCES
2nd edition, Sept. 1984 \$15/16/18 _____
An excellent source of references to articles about Forth throughout microcomputer literature. Over 1300 references.

MISCELLANEOUS

NEW FIG T-SHIRT!

601 - T-SHIRT SIZE _____

Small, Medium, Large and Extra-Large. White design on a dark blue shirt.

\$10/11/12 _____



602 - POSTER (BYTE Cover) \$15/16/18 _____

616 - HANDY REFERENCE CARD FREE _____

683 - FORTH-83 HANDY REFERENCE CARD FREE _____

FALL SPECIAL

FORTH
Dimensions

BACK VOLUMES 1-6

containing the six issues of each volume year (May - April) from 1979/80 through 1985/85.

\$50/59/90 _____

Available until November 29, 1985

PUBLICATIONS SURVEY

If you would like to suggest any other publication for review by the FIG Publications committee for inclusion in the Forth Interest Group Order Form, please complete the information below and return to FIG.

Title: _____

Author: _____

Publisher: _____

Comments: _____

Your comments on any of the publications we currently carry are most welcome, please complete information below.

Title: _____

Comments: _____

FORTH INTEREST GROUP

P.O. BOX 8231

SAN JOSE, CALIFORNIA 95155

408/277-0668

Name _____
 Company _____
 Address _____
 City _____
 State/Prov. _____ ZIP _____
 Country _____
 Phone _____

OFFICE USE ONLY		
By _____	Date _____	Type _____
Shipped By _____	Date _____	
UPS Wt. _____	Amt. _____	
USPS Wt. _____	Amt. _____	
BO Date _____	Wt. _____	Amt. _____
By _____		

ITEM #	TITLE	AUTHOR	QTY	UNIT PRICE	TOTAL
107	MEMBERSHIP				SEE BELOW

SUBTOTAL		
10% MEMBER DISCOUNT	MEMBER # _____	
CA. RESIDENTS SALES TAX		
HANDLING FEE		\$2.00
MEMBERSHIP FEE \$20/27/33	<input type="checkbox"/> NEW <input type="checkbox"/> RENEWAL	
TOTAL		

Check enclosed (payable to: **FORTH INTEREST GROUP**)
 VISA MASTERCARD
 Card # _____
 Expiration Date _____
 Signature _____

PAYMENT MUST ACCOMPANY ALL ORDERS

<p>MAIL ORDERS Send to: Forth Interest Group P.O. Box 8231 San Jose, CA 95155</p>	<p>PHONE ORDERS Call 408/277-0668 to place credit card orders or for customer service. Hours: Monday-Friday, 9am-5pm PST.</p>	<p>PRICES All orders must be prepaid. Prices are subject to change without notice. Credit card orders will be sent and billed at current prices. \$15 minimum on charge orders. Checks must be in US\$, drawn on a US Bank. A \$10 charge will be added for returned checks.</p>	<p>POSTAGE & HANDLING Prices include shipping. A \$2.00 handling fee is required with all orders.</p>	<p>SHIPPING TIME Books in stock are shipped within five days of receipt of the order. Please allow 4-6 weeks for out-of-stock books (delivery in most cases will be much sooner).</p>	<p>SALES TAX California deliveries add 6%. San Francisco Bay Area add 7%.</p>
--	---	--	---	---	---

FIG ITALIEN

Aus einem Interview mit
Marco Tarussel

Die italienische Gruppe ist vor drei Jahren entstanden. Sie hat jetzt etwa 170 Mitglieder im ganzen Land. In den größeren Städten gibt es Gruppen mit 10 bis 20 Teilnehmern, so in Mailand und in Rom. In Süditalien gibt es weniger Mitglieder.

Die FIG Italien druckt FORTH DIMENSION ab und fügt ein 24 Seiten starkes FORTH MAGAZIN mit Beiträgen in italienischer Sprache hinzu. Die Mitglieder erhalten so pro Jahr 6 Ausgaben der FORTH Dimension und des italienischen Magazines. Die englischsprachigen Beiträge werden nicht übersetzt, sondern kopiert. Beiträge aus Frankreich werden gelegentlich übersetzt. Wir stehen in Briefwechsel mit einigen FIG CHAPTER aus den USA. Deren Beiträge wurden übersetzt, angepaßt und arangiert. Der Beitrag ist 40.000 Lire im Jahr (VD: ca 22\$).

Im letzten Jahr wurden Einführungskurse in FORTH angeboten. In Milano sind regelmäßige Treffen einmal im Monat. Auch in anderen Orten finden Treffen statt, doch sind diese noch im Aufbau. In der FIG Italien ist das Verhältnis Hobbyisten zu Profis etwa 60 zu 40 geschätzt. Darunter Leute in der Lehre, an Schulen und Hochschulen. In der Universität in Parma wird FORTH gelehrt. Möglicherweise beginnt auch in Mailand ein Kurs im nächsten akademischen Jahr.

Bisher gibt es in Italien noch keine Stelle, wo FORTH-CODE systematisch gesammelt wird. Ein Problem ist auch die Kommunikation per Rechner. Der Grund dafür sind einfach die miserablen Telefonleitungen. Die Verbindungen sind sehr stark verrauscht. Modembetrieb ist da einfach nicht möglich. Drei Leute arbeiten in der Gruppe als Koordinatoren; ich selbst, Mauricio Pandolfini und Caro Schroeder. Ich selbst bin für dieses Jahr und das nächste zum Präsidenten gewählt worden.

Die Anschrift der Gruppe ist:

Marco Tarussel, FIG Italy, Via Forni 48, 2161 Milano, Italy

FORTHGRUPPE NORWEGEN

Aus einem Interview mit
David Walker

Unser erstes Forthtreffen war in Bergen, in Norwegen, im Januar 1985. Unsere Gruppe besteht aus etwa 12 Mitgliedern. Wir bemühen uns um die Anerkennung als Chapter innerhalb der Forth Interest Group. Wir sind bereits 5 Mitglieder der FIG.

Das Hauptinteresse in der Gruppe ist Process Control. Die meisten unserer Mitglieder sind von Beruf Ingenieure. Sie arbeiten in kleinen Firmen, hauptsächlich in technischen Bereichen, in denen Prototypen für den Process Control hergestellt werden. Ihr Marktbereich liegt in der Offshore Industrie, in der ölsuche und ölförderung.

Mein eigenes Interesse liegt eigentlich auf einem anderen Gebiet. Ich möchte die Benutzung von Forth in der Administration fördern. Da die Gruppe in Bergen nun hauptsächlich an anderen Dingen als ich interessiert ist, werde ich bald eine weitere Gruppe in Oslo starten und ich hoffe, daß dort im Wesentlichen administrative Programme entstehen werden.

Starkes Interesse besteht auch an Spielen für Teenager. Hier suchen wir insbesondere Multiuser Games. An Kontakten zu anderen Leuten, die Teenagern bei der Programmierung von Spielen in Forth helfen möchten, sind wir besonders interessiert. Wir denken da an Multiuser Adventures.

Die Arbeit der Leute in der Forthgesellschaft Hamburg an dem Multiuser Forth für den Commodore 64, dem Ultraforth83, finde ich äußerst interessant. Meine Tochter und etliche ihrer Freunde haben einen C64. Diese Maschine ist in Norwegen sehr populär. Von allen Teenagern, die über Computer verfügen, haben etwa 60 % den C64. Es ist die "natürliche" Maschine.

Es ist bestimmt interessant Forthwerkzeuge zu schaffen, mit denen man Spiele wie Multiuser Dungen erzeugen kann, wie es ursprünglich an der Universität von Essex entwickelt worden ist. Es müßte interessant sein, diese Spiele auf verschiedenen C64 laufen zu lassen, die miteinander kommunizieren. Dies bringt zwei Techniken in Forth ein, zum einen die Kommunikation und zum anderen das Multitasking. Eine Spracherweiterung in dieser Richtung, welche die Leute in die Lage versetzt mit komfortablen Tools Adventure Games zu erfinden, wäre sicherlich eine Möglichkeit, Forth populär zu machen. Und schließlich sind diese Techniken, welche spielerisch erfahren werden können, nicht nur für Games interessant.

Die Forthgruppe in Bergen benutzt hauptsächlich IBM-PC kompatible Rechner oder den C64. Einige der Process-control-people verwenden alle möglichen Sorten von Singleboard Computer. Ein Mitglied unserer Gruppe ist der Leiter des EDV-Departements an der Norwegian School of Economics and Business Administration in Bergen. Dort können wir DEC2060 und VAX-Systeme verwenden. In ein bis zwei Jahren werden wir wohl alle Modems haben, und Networking wird einen neuen Stellenwert bekommen. Der Austausch von

FORTHGRUPPE NORWEGEN

Aus einem Interview mit
David Walker

Publicdomaine Software für IBM-PC Kompatible und C64 ist in
vollem Gange.

Für die, die Kontakt zu unserer Gruppe in Bergen oder zu mir
aufnehmen möchten, hier meine Anschrift:

David Walker, Postbox 68, 5084 Turtnes, Norwegen.

VD

Forth ist Lisp
vom Kopf auf
die Füße gestellt.

AM LAUF DER WUPPER

von Michael Kalus

22. Juni, die Forth Interessen Gruppe im Wupper Gebiet traf sich zu ihrer zweiten Zusammenkunft im Bahnhof Ottenbruch, Funkstraße, in Wuppertal. Die Treffen dort sind jetzt regelmäßig jeden vierten Samstag im Monat ab 20.00 Uhr.

Michael Kalus berichtete über die Aktivitäten seit dem März-Treffen; der Tagungsort wurde gefunden, die zweite Nummer unseres FORTH-LOKAL herausgebracht, das F83 von Laxen/Perry ist im Tal eingetroffen. Sodann wurde das Verhältnis zur DFG und zur FIG erörtert und dem Anschluß an die Dachverbände weitgehend zugestimmt. Zum Selbstverständnis: Die Treffen sind öffentlich, kostenlos und verpflichten zu nichts. Beiträge werden nicht erhoben, jedoch darf gespendet werden. Es ist an den Teilnehmern und Forthanwendern selbst, Mitglied des Verbandes zu werden und Nutzen daraus zu ziehen. Wie, kann man bei den Treffen erfahren. Als nächstes fand dann unsere "Sprechstunde" statt, bei der fünf Fragen formuliert und die Antworten oder Wege gegeben wurden. Peter: Behandlung eines gegebenen Polynoms einer Winkelfunktion in Integer Forth?

Lothar: Hardcopy vom C64 in Forth?

Hardwig: Files contra Screens: Datenstrukturen erzeugen mit Forth. (Hier gings dann schnell hoch her, weil kein konkretes Problem bestand und demzufolge eher Weltanschauungen über Sprachen im allgemeinen ausgetauscht wurden; dennoch/deshalb packend und lehrreich.)

Marco: Compatibilität von F83 mit der Forth83 Version für Schneider PC, beide CP/M 2.2 (Vergleich demnächst auch mit ultraForth83 für C64)?

Arndt (Karlsruhe): Wie benutzt man CP/M 2.2-Files und wie den Editor im F83? (Karlsruhe hat inzwischen den Source Code zugeschickt bekommen, da mit ihrem VIEW irgend was nicht ging.).

Adolf Krüger und Dieter Lindner stellten ihre Daten-Logger-Maschine und LOLA vor, die LOGger LAnguage, geschrieben in FIG-Forth. Der Logger wurde in ein SX64 eingebaut. Adolf erläuterte die Aufgabe der Maschine und umriss die Lösung der Realtime Jobs in der Messwertverarbeitung mit Forth. Hier zeigte Forth eine starke Seite in seiner Modifizierbarkeit und in der Einfachheit Assembler Code einbinden zu können.

Es wurde die Frage aufgeworfen, ob bei solch umfangreichen CODE Passagen die gesamte Programmierung in Assembler nicht ebenso leicht und dazu kompakter und schneller sei. Forth zeigte hier jedoch bei entsprechender Faktorisierung der Worte die übersichtlichere Struktur und einfachere Handhabung des Drumherum der zeitunkritischen Teile. Die Nähe von Forth zu Assembler wurde deutlich dargestellt.

Adolf überraschte uns schließlich noch mit der Vorführung des nagelneuen ultraForth83 für Commodore C64 und SX64, welches soeben eingetroffen war. Die Graphik-Demo im Multitasking ist wirklich gut gelungen und es ist wunderbar, daß diese Forth83

AM LAUF DER WUPPER

von Michael Kalus

Version mit so ziemlich allem drum und dran Public Domain zu haben ist! Die C64 Besitzer können zufrieden sein mit der Forth Gesellschaft!

Der Rest des Abends wurde in Grüppchen weiter gefachsimpelt und es kam unser Versammlungsort, der BAHNHOF, mehr zur Geltung. Man ging über zu Bier und Mexikanischer Küche bis das Lokal geschlossen wurde.

Noch einige Bemerkungen zu unserer hiesigen Forth Gruppe. Bis jetzt sind es ca. 30 Personen die unser Journal FORTH LOKAL erhalten. Davon waren elf Leute beim letzten Treffen dabei. Unter den Forthlern befinden sich Jugendliche und Ältere, Forth-Experten ebenso wie Fortgeschrittene und Anfänger, Profis sowie Hobbyisten aus Soft- und Hardware. Doch scheint die Computerei noch eine reine Männersache zu sein. Eine aktive Forthlerin konnte bisher noch nicht begrüßt werden. Wir wollen eine offizielle Gruppe der Forth Gesellschaft (FG) und wahrscheinlich auch der Forth Interest Group in USA (FIG) werden. Dazu treffen hoffentlich recht bald genügend viele der Teilnehmer unserer Treffen in diese Organisation ein. Je fünf sind erforderlich. (übrigens ist die erste offizielle lokale Gruppe in Karlsruhe entstanden. Herzlichen Glückwunsch Leute!)

(Vierte Dimension: Inzwischen sind genügend Mitglieder aus dem Gebiet in der FG beisammen, es wurden die Koordinatoren gewählt und die lokale Gruppe ist von der FG offiziell anerkannt worden.)
VD.

FORTH GESELLSCHAFT GRUPPEN

Hamburg

Jeden vierten Samstag im Monat ab 16:00 Uhr in der
Berufsfachschule für Radio- und Fernsehtechnik,
Eimsbüttelerstr.64-66, Hamburg.
Bernd Pennemann, Eekboomkoppel 17, 2000 Hamburg 62
tel: 040-5200657

Karlsruhe

Jeden dritten Mittwoch im Monat ab 19:00 im Jugend- und
Begegnungszentrum, Krohnenplatz, Karlsruhe.
Michael Weiss, Kriegsstr.121, 7500 Karlsruhe 1
tel: 0721-854994

Paderborn

Thomas Asche, Gruningerstr.20, 4790 Paderborn
tel: 0521-26496

Wuppertal

Jeden vierten Samstag im Monat ab 20:00 im Bahnhof Ottenbruch,
Funkstr., Wuppertal-Elberfeld.
Michael Kalus, Präsidentenstr.40, 5830 Schwelm
tel: (Q) 02336-82204

Amateurfunger & Forth

Bernd Zimmermann, Werkstr.11, 2105 Seevetal 1
tel: 04105-52068

FG Gruppen in Gründung

Darmstadt

Andreas Soeder, Am Landbruch 5, 6104 Darmstadt Seeheim-Jugenheim
tel: 06257-2744

FORTH GESELLSCHAFT GRUPPEN

Berlin

Hans Madlung, Kossäten 14, 1000 Berlin 26
tel: 030-4141831

München

???

(Wir wissen, daß Ihr dort viele aktive Forthler seid! Wo ist
Eure Gruppe ?!)

Raum Frankfurt, Dreieich

Dr. Uwe Gerz, Rudolf-Binding-Weg 17, 6072 Dreieich
tel: 06103-6072

Raum Hannover, Schöningen

Eckhard Heyne, Worbergerstr.9, 3338 Schöningen 3
tel: 05352-58087

FORTH INTEREST GROUP



LESERBRIEFE

Forth und CDL auf ZX81

von Ulrich Hoffmann

CDL (Compiler Discription Language) ist eine Programmiersprache, die besonders gut für die Implementation von Compilern geeignet ist. Normalerweise läuft sie aber nur auf größeren Computern, so daß sie für Mikrocomputer-Benutzer meist nicht zur Verfügung steht.

Im Rahmen des 3. Bundeswettbewerbs Informatik reichte ich eine Arbeit über einen in Forth geschriebenen CDL/E (ein CDL Subset) - Compiler ein, der CDL/E-Programme in äquivalenten Forth-Object-Code umsetzt.

Damit ist es nun auch auf Mikrocomputern möglich, Programme in CDL zu erstellen, wobei CDL u.a. Forth's Interaktivität und Forth's gute Testmöglichkeiten gewinnt. Trotz der minimalen Hardware (ZX-81, 16K-RAM) konnte ich dabei die Leistungsfähigkeit von Forth gut ausnutzen.

Das die eingereichte Arbeit beim Bundeswettbewerb mit einem 1. Preis bewertet wurde, zeigt mir, daß ich mit Forth auf dem richtigen Weg bin. Ich bleibe jedenfalls bei Forth.

(Vierte Dimension: !!!) VD

Warum FORTH ?

Von Eckhard Heyne

Ich bin Computer Einsteiger und bei mir ist die Computerei auch ein wenig vom Geldbeutel abhängig.

Deshalb halte ich sehr viel von der Sache mit der "Public-Domain-Software" und dem Vorhaben der FORTH-Gesellschaft e.V., aktuelles Computer Wissen in die Gemeinnützung zu überführen.

Forth ist nach meinen Erkenntnissen und Informationen eine Programmiersprache mit der und an der noch lange gearbeitet werden kann. Die Idee eines Volks-Forth unabhängig von einem bestimmten System, begrüße ich, da ich nicht von den teuren Software-tools eines Programmiergiganten abhängig sein möchte, wie es zum Teil bei "BASIC" der Fall ist.

VD

LESERBRIEFE

Die ideale Programmiersprache ...

von Uwe Kloss

Es wird Zeit, daß einige "neuere Erkenntnisse" der Informatik auch auf FORTH angewendet werden! Seit ewigen Zeiten gibt es unter "Experten Streit um "die ideale Programmiersprache". Es geht dabei mehr um die Frage, ob man eine interpretierte oder kompilierte Sprache bevorzugen soll. Neuere Entwicklungen versuchen daher die Vorteile beider zu vereinen. Ich persönlich vermute, daß dabei letztendlich etwas Forth-ähnliches herauskommen wird. Allerdings nur vom Konzept her. Die anderen Programmiersprachen werden im Verlauf ihrer Entwicklungen Verbesserungen erfahren, die es ihren Verfechtern dann mit Recht erlauben, FORTH mit der Bemerkung: "Da kann ich ja gleich Assembler programmieren!" abzuqualifizieren. Der Schritt der von FORTRAN zu (z.B.) PASCAL geführt hat, ist für FORTH längst überfällig. Die Wörterbuchsuche muß beschleunigt werden (Schlüsseltransformation, auch "hashen" genannt!). Wenn das Laden von Screens länger dauert als ein Kompilerlauf, brauche ich keinen Interpreter! (Schon mal was von TURBO-PASCAL gehört?). Das Wörterbuch muß als Halde verwaltet werden. Ein "Editor", der direkt auf bereits kompilierte Worte wirkt, sollte zur Verfügung stehen. Namen sind Schall und Rauch! Das sollte mal zur Kenntnis genommen werden. FORTH ist multi-user-fähig. Namen sind user spezifisch. Also müssen Namen vom Rest des Wortes getrennt werden. Das sind beileibe nicht alle Verbesserungen, die mir so vorschweben, aber die wichtigsten. Es sind fast nur "Implementierungsdetails", die aber wegen seiner Struktur auch auf FORTH als ganzes wirken. Ich fände es großartig, wenn sich jemand aufrafft und seine Meinung zu diesem Thema hören läßt, bevor der nächste "letzte" FORTH - Standard da ist. Falls jemand seinen Ärger über diese Attacke bei mir persönlich loswerden will, hier meine Adresse:
Uwe Kloss, Vogelsang 104, 3300 Braunschweig.

(Vierte Dimension: Nur zu. Laß sehen.) VD

LESERBRIEFE

"Sitzenbleiber in FORTH"

von Klaus Haken

Vielen Dank für die Zusendung des Probeexemplars von Volume 1, Nr. 2 der "Vierten Dimension", welches ich aufmerksam durchgelesen habe. Ich begrüße die Bemühungen der Forth-Gesellschaft zur Verbreitung der Programmiersprache FORTH, insbesondere die Bestrebungen, ein VOLKSFORTH zu schaffen. Was mich allerdings derzeit noch zögern läßt Mitglied Ihrer Gesellschaft zu werden, ist das elitäre Gehabe einiger Ihrer Autoren, welche Beiträge in der oben erwähnten Ausgabe geliefert haben.

Wenn da im Kurzbericht über das Dachsberg-Treffen (S. 13, rechte Spalte, 2. Absatz) sich über Anfänger mokiert wird, welche sich mit "weltbewegenden" Themen wie COMPILE, IMMEDIATE beschäftigen oder Kurse für "Sitzenbleiber" (S.14, mittlere Spalte, 5.Absatz) offeriert werden, so mag das dem FORTH-Guru vielleicht amüsant erscheinen, für einen sich um tieferes Verständnis der FORTH-Sprache ernsthaft bemühenden Einsteiger ist das aber ein Schlag ins Gesicht.

Tatsache ist doch, daß es für den Einsteiger ziemlich schwer ist, Zugang zu einer didaktisch guten, über das nicht voll befriedigende Niveau des hochgelobten Standard-Werkes von Brody (Starting FORTH) hinausgehenden Darstellung der FORTH-Sprache zu finden. Das Buch von Zech (Die Programmiersprache FORTH) vermag meiner Ansicht nach hierzu nicht dienen, denn es scheint geschrieben worden zu sein, um FORTH-Experten die Wirkungsweise von FORTH darzustellen. Für den FORTH-Novizen ist dieses Buch kaum verständlich und daher ungeeignet.

Damit komme ich zu einer Anregung für die Gestaltung Ihrer Zeitschrift. Ich finde, Sie sollten - falls nicht schon geschehen - in Ihrer Literaturschau das Buch von Salman, Tisserand und Toulout "FORTH", ISBN 0-333-36798-7, erschienen 1984 bei MacMillan Publishers LTD, besprechen. Inhaltsangabe und Kommentar sind in Kopie als Anlage beigefügt. Ich halte dieses Buch, bei welchem u.a. auch der Leiter von FIG UK, de Grandis-Harrison, mitgewirkt hat, für das Beste, was es derzeit auf dem Büchermarkt gibt, um den Anfänger mit allen wesentlichen Eigenschaften der FORTH-Sprache didaktisch einwandfrei vertraut zu machen.

(Vierte Dimension: Ihre Schelte ist berechtigt. Entschuldigung.)
VD

BUCHBESPRECHUNGEN

von Jens Storjohan

M. Ben-Ari, Grundlagen der Parallelprogrammierung
Carl Hanser Verlag München Wien
1985 , ISBN 3-446-14155-3

Das vorliegende Buch ist eine gelungene Übersetzung aus dem Englischen. Es ist ähnlich wie die bekannten Bücher von Donald Knuth (The Art of Computer Programming) gleichzeitig sorgfältig und witzig geschrieben. Das gute Druckbild und die schönen Zeichnungen verdienen Erwähnung.

Es erklärt die grundlegenden Methoden der Parallelprogrammierung (concurrent programming). Das Hauptaugenmerk liegt dabei auf Echtzeitsystemen für Mini- und Mikrorechner.

Der Autor benutzt, soweit er Programmiersprachen zur Beschreibung der Algorithmen benötigt, eine Untermenge von Pascal und einige Elemente von Ada. Trotzdem sollte auch der Forth-Anhänger sich nicht davor scheuen, die wesentlichen Ideen zwischen den vielen BEGIN'S und END'S des Pascal-Textes zu suchen.

Wichtig sind immer die Beweise für die Fehlerfreiheit der Algorithmen, die in einem halbformalen Stil gehalten und dadurch recht gut verständlich sind. Dieser Zug der Entwicklung der Informatik, nicht an die Fehlerfreiheit von Programmen zu glauben, sondern sie zu beweisen, spielt in der Echtzeitprogrammierung, in der es schwer ist "alle Möglichkeiten durchzuspielen", eine besondere Rolle.

Die Kapitelüberschriften lauten:
Was ist paralleles Programmieren?
Abstraktionen der Parallelprogrammierung.
Das Problem des gegenseitigen Ausschlusses.
Semaphore.
Monitore.
Das Rendezvous in Ada.
Die speisenden Philosophen.
Anhang: Implementierungsbausatz.
Literaturangaben.
Stichwortverzeichnis.

Insgesamt ein empfehlenswertes Buch für alle, die sich mit Echtzeitprogrammierung befassen. VD

BUCHBESPRECHUNGEN

von Jens Storjohan

Terry E. Shoup
Numerische Verfahren für Arbeitsplatzrechner
Carl Hanser Verlag München Wien 1985
ISBN 3-446-141182-0

Das vorliegende Buch beschreibt numerische Verfahren und liefert fertige BASIC-Programme. Die Kapitel befassen sich, ausser einer Einführung über Rechner und Mikrocomputer, mit den Themen algebraische und transzendente Gleichungen, Lösung von Gleichungssystemen, Eigenwertprobleme, gewöhnliche Differentialgleichungen, Interpolation und Kurvenanpassung, numerische Differentiation und Integration.

Es setzt eine gewisse mathematische Vorbildung voraus, wie sie bei Fachhochschul-Ingenieuren oder Studenten, die Mathematik im Nebenfach betreiben, erwartet werden kann. Teile des Buches sind auch mit geringeren Vorkenntnissen zu verstehen.

Die vorgestellten Verfahren stellen eine sinnvolle Auswahl dar. Die Erklärung der Wirkungsweise der Verfahren ist gut. Dies liegt nicht zuletzt daran, daß der Verlag nicht an Abbildungen gespart hat. Das Druckbild ist ausgezeichnet. Die Übersetzung ist fachmännisch durchgeführt. Der Übersetzer hat das Buch auch noch durch gut ausgewählte Literaturhinweise in deutsche Sprache ergänzt.

Es gibt so viele Bücher mit Anwenderprogrammen in BASIC, die triviale Programme mit hingesudelten Erklärungen enthalten, daß man die Ausnahmen, zu denen dieses Buch gehört, begrüßen sollte. Meine Kritik zielt im wesentlichen auf drei Punkte.

a) Die Einführung über Rechner, ihre Architektur und Wirkungsweise erscheint mit überflüssig. Ein zusätzlicher Abschnitt z. B. über Optimierung wäre nützlicher gewesen.
b) Das verwendete BASIC läßt keine strukturierte Programmierung zu. Für den Forth-Programmierer bedeutet dies, daß er den Beispielpogrammen erst einmal eine Struktur aufprägen muß, bevor er sie in ein Forth-Programm umsetzt. Eine Sammlung von Fortran-Unterprogrammen eignet sich da besser zur Übersetzung in eine strukturierte Sprache, weil wenigstens eine Modularisierung und solche Hilfsmittel wie lokale Variable zur Verfügung stehen. Viele moderne Varianten vom BASIC unterstützen lokale Variable, echte Prozeduren usw., wenn auch nicht auf eine genormte und kompatible Weise. Aber man kann es dem Autor nicht vorwerfen, wenn er seine Programme für einen verbreiteten Rechner (Apple II) schreibt.

c) Ein weiterer Punkt der Kritik bedarf einiger erklärender Vorreden. In der numerischen Mathematik ist es sehr viel einfacher, den Ablauf eines Verfahrens zu beschreiben als präzise die Qualität und den Anwendungsbereich. Zum Beispiel läßt sich das übliche Verfahren zum "Wurzelziehen mit der Hand"

BUCHBESPRECHUNGEN

von Jens Storjohan

Mittelstufenschülern beibringen, aber die Frage, wieviele Durchläufe man zur Erreichung einer vorgegebenen Genauigkeit benötigt, ist schwieriger zu klären.

Bei den Verfahren, die in dem Buch vorgestellt werden, ließen sich solche Fragen nur mit erheblich größeren mathematischen Kenntnissen formulieren und beantworten, als sie der Autor und die Leser, an die er sich wendet, haben. Der Autor hat sich bemüht, dem Leser Hinweise zu geben, mit denen er das im Einzelfall geeignetste Verfahren auswählen kann. Dies ist ihm im Abschnitt über Differentialgleichungen ganz gut gelungen. In den anderen Abschnitten fällt es dem Fachmann störend auf, daß der Autor zu sehr die Benutzung auch einfacher mathematischer Begriffe vermeidet und nicht genügend eigene Erfahrungen ins Spiel bringt.

Leider versäumt er es z. B. bei den Verfahren der linearen Algebra quantitative Angaben über den Aufwand an Rechenoperationen und damit an Rechenzeit, in Abhängigkeit von den Dimensionen der auftretenden Matrizen, zu geben. Dies wäre mathematisch nicht aufwendig gewesen und auch für den Praktiker nützlich. Ebenso macht er im Abschnitt über die Lösung von Gleichungssystemen irreführende und unsinnige Bemerkungen über den Konvergenzbereich beim allgemeinen Iterationsverfahren und Newton-Verfahren.

Für den typischen Benutzer dieses Buches ist dies vermutlich kein schwerwiegender Mangel, denn er wird ohnehin mehr dazu neigen, passende Verfahren durch probieren zu finden, so daß ihm meine mathematische Mäkelei übertrieben vorkommen mag.

Alles in allem handelt es sich doch um eine ordentliche einführende Beschreibung der modernen Standardverfahren der Numerik, die man empfehlen kann. VD-

Protokoll

der Mitgliederversammlung
der Forth Gesellschaft
am 24.11.85 in Bergzabern.

1. Eröffnung, Festlegung der Tagesordnung

Die versammelten Mitglieder wählen Michael Kalus zum Versammlungsleiter. Dieser eröffnet die Versammlung und stellt fest, daß die Beschlußfähigkeit gegeben ist, da 21 Mitglieder anwesend sind. Die Tagesordnung wird wie folgt beschlossen:

1. Eröffnung
2. Geschäftsbericht des Direktoriums
3. Berichte aus den lokalen Gruppen
4. Haushaltsplan
5. Geplante Satzungsänderung
6. Wahl des Direktoriums
7. Verschiedenes

2. Geschäftsbericht des Direktoriums

Zunächst berichtet H.G. Lynsche über die finanzielle Lage der Gesellschaft. Der Kontostand betrug am 7.11.85 DM11720,47. Für das laufende Jahr sind noch die folgenden Kosten zu erwarten: DM2000 für die Vierte Dimension, DM500 für den Steuerberater, DM800 für Verwaltungskosten. Diese recht günstige Finanzlage ist hauptsächlich den Einnahmen aus der FORML-Konferenz auf Burg Stettenfels zu verdanken.

Jens Storjohan berichtet über die Arbeit des Direktoriums an der Aufbauphase des Vereins. Die Gemeinnützigkeit wurde erreicht. Es gilt nun diese verstärkt wahrzunehmen.

Klaus Schleisiek schilderte sodann den etwas verworrenen Rechtsstreit mit der ForthInc über die Verwendung des Ausdrucks FORTH und berichtet von der EuroForml-Konferenz.

Schließlich werden Aktenordner mit Kopien der in der Zentrale gesammelten Literatur an die Vertreter der lokalen Gruppen aus Wuppertal, Paderborn, Hamburg und Karlsruhe übergeben.

Anschließende Diskussion. Hauptpunkte der Kritik waren die als zu hoch angesehenen Verwaltungskosten und das Ausbleiben des Forth Magazines "Vierte Dimension". Dem Direktorium wurde die Entlastung erteilt mit 20 Ja, 00 nein und 03 Enthaltungen.

3. Berichte aus den lokalen Gruppen

Paderborn:

Thomas Asche als lokaler Koordinator und Bernfried Molte organisieren hier ein monatliches Treffen.

Hamburg:

An Stelle des hier abwesenden lokalen Koordinators Bernd Pennemann berichtet Karsten Röderer über die Aktivitäten der Gruppe.

Protokoll

der Mitgliederversammlung
der Forth Gesellschaft
am 24.11.85 in Bergzabern.

Wuppertal:

Michael Kalus berichte über die Gruppe im Wuppertal. Sie geben eine eigene Zeitschrift heraus, die FORTH LOKAL.

Berlin:

Hans Madlung übernimmt es, sich um die Organisation der Berliner Forthanhänger zu kümmern.

Karlsruhe:

Michael Weiss berichtet über die Gruppe und ihre Aktivitäten. Sie haben diese Jahreshauptversammlung ausgerichtet. Ihnen gebührt der historische Ruhm, die erste lokale Gruppe der Forth Gesellschaft gegründet zu haben.

Darmstadt:

Andreas Soeder berichtet über Forthaktivitäten, die sich um einen von ihm geleiteten Volkshochschulkurs über Forth bilden. Die Versammlung hofft, daß daraus eine lokale Gruppe entstehen wird.

Im Anschluß hieran stellte die Mitgliederversammlung einstimmig fest, das der Kriesenrat zur Zeit aus den Mitgliedern der folgenden lokalen Gruppen gebildet wird:
Karlsruhe, Wuppertal, Paderborn und Hamburg.

4. Haushaltsplan

K.Schleisiek gibt eine Begründung der Pläne eine eigene Mailbox einzurichten. Die geschätzten Kosten betragen DM47 monatliche Postgebühren und als einmalige Ausgaben ca DM4000 bis 5000 für einen Rechner (PC20) und DM1500 für ein Telemodem. H.G.Lynsche schätzt, das nach dem bisher üblichen Abrechnungsverfahren die Kosten für die Verwaltung maximal DM6000 betragen werden. Herr Klingenberg stellt den Antrag, die Verwaltungspauschale in einen Sockelbetrag und einen verminderten Proportionalbetrag entsprechend der Mitgliederzahlen zu unterteilen. Der Antrag wird angenommen mit 7 ja, 6 nein und 8 Enthaltungen. K.Schleisiek beantragt, dem Projekt Mailbox für das Haushaltsjahr 1986 DM8000 zur Verfügung zu stellen. Der Antrag wird angenommen mit 21 ja, 1 nein und 1 Enthaltung.

5. Satzungsänderung

Der Antrag, die Wahlperiode des Direktoriums auf zwei Jahre auszudehnen, findet mit 12 ja, 9 nein und 2 Enthaltungen nicht die erforderliche 2/3 Mehrheit.

Protokoll

der Mitgliederversammlung
der Forth Gesellschaft
am 24.11.85 in Bergzabern.

6. Wahl des Direktoriums

Klaus Schleisiek, H.G.Lynsche und Jens Storjohan stellen sich als Kandidaten zur Verfügung. Auf Vorschlag des Versammlungsleiters hin wird per Handzeichen abgestimmt. Die genannten Kandidaten werden mit 23 ja-Stimmen einstimmig gewählt.

7. Verschiedenes

Es wurde beantragt, die Mitgliedserklärung so zu erweitern, daß abgefragt wird, welche persönlichen Informationen nur dem Vorstand, nur innerhalb des Vereins oder auch Interessenten außerhalb des Vereins mitgeteilt werden dürfen. Angenommen mit 12 ja, 4 nein, 4 Enthaltungen. Der Antrag, zusammen mit der nächsten Vierten Dimension (Vol.II, Nr.1 1986) eine Mitgliederliste zu versenden, wird mit 16 ja, 1 nein und 5 Enthaltungen angenommen. M.Kalus berichtet, daß er die Erstellung der Vierten Dimension übernehmen wird. Die Zeitung soll aber weiterhin in Hamburg vorbereitet, gedruckt und verschickt werden. K.Schleisiek wird eine Gruppe von Experten zusammen bringen, die den eingesandten Forthcode auf Richtigkeit und Wertigkeit für die Veröffentlichung prüfen soll.

Die Versammlung spricht der lokalen Gruppe Karlsruhe Ihren Dank für die gelungene Organisation aus.

Finanzamt für Körperschaften in Hamburg
Geschäftszeichen (bei Schreiben und Überweisungen bitte angeben)

23/444/03358

Finanzamt für Körperschaften in Hamburg, Postfach 10 22 42
2000 Hamburg 1

Hamburg 1, 14. 10. 1985

Wendenstraße 10
Zimmer 413

Postnetz - Durchwahl - Vermittlung - Behördennetz
2 57-96-7 2730 (-1) 9.53-

Forth Gesellschaft e.V.
c/o Herrn Dr. Bernd Müller
Lüneburger Str. 24
2100 Hamburg 90

Vorläufige Bescheinigung

Für den Empfang steuerbegünstigter Spenden wird hiermit vorläufig bescheinigt, daß die o.a. Körperschaft nach der eingereichten Satzung durch Förderung der Volksbildung auf dem Gebiet der Programmiersprache Forth

ausschließlich und unmittelbar gemeinnützige mildtätige kirchliche religiöse wissenschaftliche Zwecke verfolgt und zu den in § 5 Abs. 1 Nr. 9 KStG bezeichneten Körperschaften, Personenvereinigungen und Vermögensmassen gehört.

Die satzungsmäßig verfolgten gemeinnützigen Zwecke sind nach Nr.5.... der Liste in Anlage 7 zu den Einkommensteuer-Richtlinien (EStR) allgemein als besonders förderungswürdig im Sinne von § 10 b Einkommensteuergesetz bzw. § 9 Nr. 3 a Körperschaftsteuergesetz anerkannt. +)

Die Körperschaft ist berechtigt, für ihr unmittelbar zugewendete Spenden Bescheinigungen nach dem anliegenden Muster zu erteilen.

Diese Bescheinigung ist widerruflich. Sie bindet weder die für die Veranlagung der Spender zuständigen Finanzämter noch stellt sie eine Entscheidung über die zukünftige steuerliche Behandlung der Körperschaft selbst dar.

Diese Bescheinigung verliert ihre Gültigkeit, sobald ein Körperschaftsteuerbescheid bzw. ein Freistellungsbescheid für die o.a. Körperschaft ergangen ist. Sie gilt längstens 18 Monate vom Ausstellungsdatum ab gerechnet.

Unterschrift

Isenbart

Das Zutreffende ist angekreuzt

+) Nichtzutreffender Absatz ist gestrichen

FORTH GESELLSCHAFT EV

BÜRO: SCHANZENSTRASSE 27

2000 HAMBURG 6

TEL: 040 - 43 50 70 (Mo: 11-18 Uhr)

ANTRAG

Ich möchte der FORTH Gesellschaft eV beitreten. Dafür erhalte ich die VIERTE DIMENSION (4 Ausgaben pro Jahr) frei Haus .

Mein Beitrag soll:

- () --- DM 32,- / Jahr (Stud./Arbeitsl. mit Ausweis)
- () --- DM 64,- / Jahr (Ordentliches Mitglied)
- () --- DM 128,- /Jahr (Förderndes Mitglied=Firmen/Institutionen)

Ausserdem unterstütze ich die Forth Gesellschaft eV mit einer Spende von _____ DM.

NAME: _____

Strasse: _____

PLZ/ORT: _____

TEL: _____ / _____

UNTERSTÜTZUNG:

() - Ich möchte eine Gebietsgruppe gründen. Daher bin ich mit Veröffentlichung meiner Adresse in der VIERTE DIMENSION einverstanden. Ich bin bereit aktiv an der Entscheidungsfindung der Forth Gesellschaft eV teilzunehmen.

() - Ich möchte eine Fachgruppe gründen . Daher bin ich mit der Veröffentlichung meiner Adresse in der VIERTE DIMENSION einverstanden. Ich bin bereit aktiv an der Entscheidungsfindung der Forth Gesellschaft eV teilzunehmen. Das Thema der Fachgruppe soll sein:

() - Ich möchte die Forth Gesellschaft eV unterstützen in dem ich Artikel / Buchbesprechungen / Produktnews für die VIERTE DIMENSION schreibe resp. meine bisherigen Artikel etc. an die Redaktion sende. Das Copyright für diese Artikel liegt bei mir.

() - ich habe ausserdem noch folgende Anregungen / Kritiken :

Den Gesamtbetrag von DM _____ , - habe ich am _____
auf das Konto (siehe unten) überwiesen ()
als Verrechnungsscheck beigelegt ()

DATUM & ORT : _____ Unterschrift: _____

SPENDEN- & BEITRAGSKONTO: FORTH GESELLSCHAFT EV

POSTGIROAMT HAMBURG, (BLZ: 20010020), KTO: 56 32 11 - 208