

Hans-Georg Joegen

Focal auf dem Vormarsch

2. Teil

Nach dem Ausflug in die Geschichte und einem Blick auf die Struktur von FOCAL-65 setzen wir unsere Betrachtung des neuen Interpreters für 6502-Mikrocomputer mit der Erläuterung spezieller Eigenheiten der Sprache und dem Bericht über Betriebserfahrungen auf einem Computer ITT-2020 fort. Der Interpreter wird als Kassette für verschiedene Computerfabrikate von der Software-Börse „The 6502 Program Exchange“ in Reno vertrieben.

Im folgenden wird mehrfach auf die im letzten Heft erschienene Tabelle 1, „FOCAL-Funktionen“, Bezug genommen; es ist empfehlenswert, sie zur Hand zu nehmen. Anstatt der schönen deutschsprachigen Kommandoworte, die Programmier-Anfängern aus unseren Breiten die erste Bekanntschaft mit dieser zauberhaften Programmiersprache so erleichtern, benutzen wir fortan die international üblichen Kurzzeichen aus dieser Tabelle. Dies gilt bereits für das Beispiel-Programm in Bild 4, dem zur Kontrastbildung und Gegensatz-Erläuterung ein Parallelprogramm in Basic gegenübergestellt ist (Bild 5). Dieses Programm-Pärchen macht schön deutlich, daß Programmieren in Focal mehr ist als Programmieren in Basic auf

prägnantere Form: Was die drei Basic-Zeilen 2002 bis 2004 tun, erledigt in Focal ein einziges Statement in Gruppe 20, Zeile 02. Ähnlich wie das bereits besprochene I-Kommando („INSO-FERN“ oder „IF“) einen Klammersausdruck auswertet und danach zu einer von drei aufgeführten Zeilen springt, benutzt O „Operator-Routine“, „On“) die drei Fälle Klammersausdruck kleiner, gleich oder größer Null zum Unterprogramm-Aufruf. Wir werden später sehen, zu welcher eleganten Problemlösung derlei verhilft: Ein Fortran-Erbe, das sich auch andere Hochsprachen zunutze machen. Augenfalliger noch wird die gegenüber Basic und erst recht im Vergleich mit Pascal sichtbare Programm-Prägnanz im

Beispiel aus Bild 6. Zugrunde liegt ein Schleifchen, mit dessen Hilfe Anfang 1980 in der FUNKSCHAU auf die „Null-Probleme binärer Basic-Interpreter“ aufmerksam gemacht wurde [1]. Nebenbei bemerkt: Auch Focal purzelt in die binäre Null-Falle und zeigt einen Fehler gleicher Größenordnung wie die damals erwähnten Basic-Interpreter, was der Probelauf verrät. Aber darauf kommt es in diesem Zusammenhang nicht an: Gezeigt werden soll, wie lakonisch sich mit Hilfe des Folge-Kommandos F eine solche Schleife bauen läßt. In Basic müßte man bekanntlich schreiben „FOR N=-1 TO 1 STEP -1: GOSUB 30“. Und was den zweiten Demonstrations-Sachverhalt aus Bild 6 angeht, die bequeme Ausgabe-Formatierung einschließlich eingebauter Rundung mit Hilfe des Prozentzeichens und nachfolgender Zifferangabe (Vorkomma-Stellen/Punkt/Nachkommastellen): Da gibt es unterdessen mehr als ein Dutzend Veröffentlichungen diesseits und jenseits des Atlantik, die Tricks und Winkelzüge nenne, mit deren Hilfe man sich derlei auf teilweise recht komplizierte Weise in seine PET-Programme erst einmal einzubauen hat – was gleichermaßen auch für Applesoft und Palssoft gilt, lediglich das Level-II des TRS-80 wartet hier (Print Using) mit vergleichbarem Komfort auf.

„BASE“: Voller Griff in die Trickkiste

Die bisher vorgestellten Musterprogrammchen hatten einen einzigen Zweck: Auf spezielle Focal-Eigenheiten aufmerksam zu machen und Unterschiede zu Basic zu verdeutlichen. Nun betrachten wir ein Nutzprogramm hoher

```

JM
C FOCAL-65 (V3E) 1-0CT-79
20 01 A "TESTZAHL? ",A
20 02 O (A-100)21,22,23
20 03 T " HUNDERT -",!G 20,01
21 01 T "KLEINER ALS "
22 01 T "GLEICH GROSS WIE "
23 01 T "GROSSER ALS "

LD
TESTZAHL? 11
KLEINER ALS HUNDERT. -
TESTZAHL? 111
GROSSER ALS HUNDERT. -
TESTZAHL? 100
GLEICH GROSS WIE HUNDERT. -
    
```

Bild 4. Eine Mini-Aufgabe in Focal

```

LIST
2001 INPUT "TESTZAHL? ";A
2002 IF A < 100 THEN GOSUB 2101
2003 IF A = 100 THEN GOSUB 2201
2004 IF A > 100 THEN GOSUB 2301
2005 PRINT "HUNDERT -" :GOTO 2001
2101 PRINT "KLEINER ALS " : RETURN
2201 PRINT "GLEICH GROSS WIE " : RETURN
2301 PRINT "GROSSER ALS " : RETURN

RUN

TESTZAHL? 11
KLEINER ALS HUNDERT. -
TESTZAHL? 111
GROSSER ALS HUNDERT. -
TESTZAHL? 100
GLEICH GROSS WIE HUNDERT. -
    
```

Bild 5. Die Mini-Aufgabe aus Bild 4, diesmal – zum Vergleich – in Basic formuliert

Leistungsfähigkeit, das Zwei-Operanden-Mathematik in zwei beliebigen Zahlensystemen ausführt und in einem dritten System antwortet („Was kommt als Oktalzah raus, wenn ich die Hexzahl FECD durch die Binärzahl 11011 teile?“). Das Programm (Bild 7) reserviert zu Beginn Speicherraum von zweimal 16 Nutzbyte, auf die fortan über die Namen AS und BS zugegriffen wird. (Funktion FISL, „Function Initialize String Length“.)

Die Folgezeile gewinnt aus B1 mit Hilfe von Gruppe 3 (gerufen durch FSBR) einen Kontrollzähler D1, der angibt, wieviel Ziffern BASE an dieser Stelle maximal handhaben kann. In 1.30 wird die Variable O mit dem ASCII-Code des Operators geladen, beispielsweise 43 für ein Pluszeichen. Wie sich nun der „BASE-Algorithmus“ im einzelnen weiterentfaltet, kann man sehr schön weiterverfolgen, wenn man einen Bleistift zur Hand nimmt und von hier ab den Programmweg selbst weiterverfolgt – man gewinnt bei derlei Analysieren mehr an „Focal-Feeling“ als durch passive Rezeption. Soviel Hilfe sei dem Leser jedoch, außer den beiden zum Programmverständnis unerlässlichen Tabellen, zusätzlich mit auf den Weg gegeben: In Zeile 1.50 wird die Kette C\$ mit der Anweisung (in ASCII formuliert) „S C=A“ vorgeladen, um das Operatorzeichen vermehrt, das der Benutzer eingab, und mit dem zweiten Operanden B ergänzt. C\$ war zum gültigen Ausgabekanal erklärt worden; R O („Rückstellen Outputkanal“) schaltet auf den Schirm zurück.

Die folgenden Zeichen D C\$ („Durchführung C-String“) haben es in sich: Der Interpreter koppelt den Rechner vom laufenden Programm ab und beginnt nun, die Anweisungen in C\$ abzuarbeiten! Mit anderen Worten: Ein durch Software simulierter und damit, bei Licht betrachtet, virtueller Datenkanal übernimmt vorübergehend das Kommando über das Gesamtsystem – solange, bis das Zeichen CR gelesen wird. Mit diesen Hinweisen zu dem einer meisterhaften Programmierer-Feder entstammenden „BASE“ wollen wir die allgemeine Betrachtung dieser jüngsten Version von Focal beschließen und uns dem konkreten Interpreter zuwenden, den man für seine 30 Dollar bekommt. Es gibt Versionen für den KIM und eine Apple-Fassung, die wir auf einer ITT-2020 erproben. Darüber hinaus wird eine Fassung für den Monitor TIM angeboten, wie er für ältere PETs als Band zu haben und in neueren CBM-Maschinen bereits resident eingebaut ist.

Konkrete Betriebserfahrungen

Die Kassette lud einwandfrei in den Speicherbereich \$ 4000...\$ 6001 unserer 48-KByte-Maschine. Darauf verlief ein anscheinend ungestörter Start, später stellte sich jedoch heraus, daß die Verbindung zum Disketten-Betriebssystem DOS 3.2 verloren war. Die in Tabelle 3 aufgeführten Änderungen, vorgenommen vom Monitor der 2020 aus, beseitigten diesen Mangel. Anweisungen an DOS 3.2 wurden fortan ausgeführt, wenn dem Kommando ein ASCII-Sonderzeichen mit dem Wert 4 (Control-D) vorangestellt wurde. Programme, die mit Hilfe eine Editors konzipiert, als Textfile auf Diskette gespeichert und dann von Focal aus mit „EXEC (Name)“ aufgerufen wurden, luden einwandfrei, ohne zuvor im Speicher vorhandene Programme zu stören, wenn diese andere Gruppennummern verwendeten als das Folgeprogramm. Laden und Abspeichern ohne Umwege dagegen ist kompliziert: Man muß, vom Monitor aus, den Pointer \$ 93, \$ 94 abfragen und findet so das Programm-Textende. Anfang ist immer \$ 6002.

Mit Hilfe der so gewonnenen Information läßt sich ein binärer Speicherbefehl BSAVE erteilen. Die Rückholung erfolgt hinwiederum bequem durch BLOAD (Name). Eine neue Focal-Fassung für Diskettenbetrieb ist angekündigt; wir erstanden die anfänglich allein erhältliche Kassettenversion, so daß Wayne Wall und seinen Freunden wegen der Disketten-Unbequemlichkeiten kein Vorwurf zu machen ist: Schön, daß sich Focal nun mit DOS 3.2 überhaupt verträgt! Als hilfsbereit zeigte sich beim Einbinden des Interpreters in das Diskettenbetriebssystem übrigens Don Marshal von der „6502 Program Exchange“ – in zwei nächtlichen Transatlantik-Telefongesprächen gab Don bereitwillig und engagiert Auskünfte zu solchen Fragen, in denen die (gesondert zu bestellenden) Dokumentationsmaterialien [6; 7] den Leser allein lassen. Derlei Lücken in der Dokumentation gibt es leider mehrere: Nirgendwo wird auf die komfortablen Eingabemöglichkeiten verwiesen, die der Verfasser durch Zufall entdeckte: Variable und Ausdrücke als Antwort auf „ANFRAGE“. Als Schönheitsfehler am Interpreter selbst zeigte sich weiter, daß er als Meldezeichen ein Sternchen verwendet – justament das gleiche Zeichen, das auch der Apple-Monitor benutzt, um anzukündigen, daß er zur Stelle sei. Wir bauten den Tiefstrich als Focal-Prompting-Character ein, die Änderung ist in den Angaben der Tabelle 3 enthalten.

**Präzisionstest „Kommissar“:
Mit Glanz und Glorie bestanden**

Den Präzisionstest „Kommissar“ [2] bestand FOCAL-65 mit Glanz. Nach 861 Iterationen lieferte der Interpreter einen Wert für die Eulersche Zahl, die nur um $580 \cdot 10^{-6}$ vom wahren neunstelligen Wert abwich. Freilich benötigte die ITT-2020 dafür rund fünfmal mehr Zeit, als hätte sie unter Palsoft gerechnet. Zu den Schnellsten im Lande zählt der neue Interpreter damit durchaus nicht. Als unschön empfindet der Verfasser weiter, daß beim Interpreter zwar erfolgreich mit jedem Byte gezeitet wurde, daß der Programmtext selbst sich aber unverdichtet als ASCII-Kette im Speicher breitmacht [10]. Freilich bringt das wieder den Vorteil der bequemen Editierung: Die mit MODIFIZIEREN aufgerufenen Dienstleistungen erlauben angenehme Änderungsarbeit; allerdings vermißt man schmerzlich ein noch nicht erhältliches Renummerierungs-Paket. Dieses – vielleicht sogar in Focal – selbst zu schreiben, dürfte jedoch nicht zu schwer sein, da die Organisationsstruktur des abgelegten Programmtexes keine Geheimnisse birgt: Einschließlich der Ziffern wörtlich ASCII mit den Zeichen \$ F und \$ FF als Programmende-Markierung.

```

-JW
C FOCAL-65 (V3E) 1-OCT-79
10.10 C
10.20 C FALLENTEST
10.30 C -----
10.40 C
20.10 F N=-1, 1,1:D 30
20.20 0
30.10 T % 2.09.N " ", % 2.08." "N
30.20 T % 2.07." "N. !

-D
-1.000000000 -1.000000000 -1.000000000
-0.900000000 -0.900000000 -0.900000000
-0.800000000 -0.800000000 -0.800000000
-0.700000001 -0.700000000 -0.700000000
-0.600000001 -0.600000000 -0.600000000
-0.500000002 -0.500000000 -0.500000000
-0.400000002 -0.400000000 -0.400000000
-0.300000002 -0.300000000 -0.300000000
-0.200000002 -0.200000000 -0.200000000
-0.100000002 -0.100000000 -0.100000000
-0.000000002 -0.000000000 -0.000000000
0.099999997 0.100000000 0.100000000
0.199999997 0.200000000 0.200000000
0.299999997 0.300000000 0.300000000
0.399999997 0.400000000 0.400000000
0.499999997 0.500000000 0.500000000
0.599999997 0.600000000 0.600000000
0.699999996 0.700000000 0.700000000
0.799999996 0.800000000 0.800000000
0.899999995 0.900000000 0.900000000
0.999999995 0.999999999 1.000000000
    
```

Bild 6. Die Möglichkeit zu prägnanten Formulierungen und eine Routine, die Zahlen sauber formatiert ausgibt und auch rundet – Beispiele für die zahlreichen Vorzüge dieser höheren Programmiersprachen. Beim „Fallentest“ freilich versagt auch Focal

FOCAL-65 – für wen?

Die weiten Freiräume, die der neue Interpreter dem Programmierer einräumt, legen den Schluß nahe, bei FOCAL-65 handle es sich um eine für den Anfänger weniger geeignete Sprache. Genau das Gegenteil ist richtig! Die Möglichkeit, anfänglich mit voll ausgeschriebenen, in Deutsch oder Englisch formulierten Anweisungen zu arbeiten, erleichtert das Erlernen der Sprache ungemein. Dies gilt besonders für jene Zeitgenossen, die mit dem Englischen nicht gerade auf Du stehen und nicht von vornherein feste Vorstellungen mit Texten wie IF A = 1 THEN FOR N = A TO B STEP C verbinden. So betrachtet, erscheint FOCAL-65 als eine beispielsweise für deutschsprachige Schulen vorzüglich geeignete Programmier-Sprache. Dem steht freilich entgegen, daß es nur wenig fertige Software für FOCAL-65 gibt, und inwieweit die in früheren Versionen dieser Sprache entwickelten Programme kompatibel sind, konnte lediglich aus Literaturrecherchen positiv beantwortet werden, praktische Versuche dazu fanden nicht statt.

Schlußurteil: Wem die Langatmigkeit von Basic auf die Nerven geht und wer der noch größeren Geschwätzigkeit von Pascal keinen Vorteil abgewinnen mag: Für den ist das frischverjüngte Focal eine interessante Alternative. Leider gibt es zur Zeit noch keinen europäischen Händler für FOCAL-65. Man muß also direkt in den Staaten bestellen [8], will man nicht auf „schwarze“ Kopien hoffen – und das sollte man weder Wayne Wall noch Don Marshal antun, denn die haben der Mikrocomputer-Branche durch gute Arbeit einen vorzüglichen Dienst erwiesen.

Spruch des Monats

Mit der Mechanisierung schematischer geistiger Arbeiten auf breiter Grundlage beginnt ein neuer Abschnitt der Technik. Die in diesem Zusammenhang auftretenden Probleme sind so umfangreich, daß eine Generation von Wissenschaftlern, Technikern und Wirtschaftlern erforderlich sein wird, um sie erschöpfend zu bearbeiten. Da die Entwicklung der beschriebenen Rechengereäte ihre umfassendsten Anwen-

```
C FOCAL-65 (V3E) 1-0CT-79
1.01 C "BASE" FOR FCL65E COPYRIGHT 1977 LARRY FISH
1.02 C ARITHMETIC IN BASES 2 THROUGH 36 (DIGITS 0-2)
1.03 C NEGATIVE NUMBERS NOT ALLOWED ON INPUT
1.05 E
1.10 S F1SL(16,A#,16,B#):A "BASE OF FIRST NUMBER: ",B1
1.20 S D1=FSBR(3,B1):T "FIRST NUMBER: ",S J=FSTI(D1,A#,13):T !
1.25 A "BASE OF SECOND NUMBER: ",B2: S D2=FSBR(3,B2)
1.27 T "SECOND NUMBER: ",S J1=FSTI(D2,B#,13):T !
1.30 T "OPERATION: ",S O=FCHR():A "BASE OF OUTPUT: ",B3
1.40 S B4=B1,D=J,N=0: D 2.1: S A=Z,B4=B2,D=J1,N= 0:1: D 2.1: S B=Z
1.50 S FODV(C#):T "S C=A":S FOUT(O):T "B",:R 0: D C#
1.60 D 4

2.10 S Z=0:F X=0,D=D-1: D (2.3+N): I (Y-64),, 2.2: S Z=Z+(Y-48)+B4*(D-X)
2.20 S Z=Z+(Y-55)+B4*(D-X)
2.30 S Y=A#(X)
2.31 S Y=B#(X)

3.10 I (&-4),, 3.5: D (3.2+8/100): R
3.22 S &=16
3.23 S &=11
3.24 S &=8
3.50 I (&-15),, 3.9: I (&-9),, 3.8: I (&-6): 6.3.6.3.7
3.60 S &=7: R
3.70 S &=6: R
3.80 S &=5: R
3.90 S &=4: R

4.10 I (B3-10), 4.35: D (C)4.5: S D=FSBR(3,B3)
4.15 S D=D-1
4.20 I (D)4.4: S T=FINT(C/B3^D),C=C-(B3^D*T): I (T-9),, 4.3: T %: T) G 4.15
4.30 S FOUT(T+55): G 4.15
4.35 T C
4.40 T !, 0
4.50 S C=C-T " "-
```

Bild 7. Das kommt heraus, wenn Profis in Focal schreiben – ein Programm, das Berechnungen simuliert in drei verschiedenen Zahlensystemen durchführt

Tabelle 3: Vorgeschlagene Änderungen am Interpreter und an DOS 3.2 (alle Angaben in Hexnotation)

Speicherzelle	Alter Inhalt	Neuer Inhalt
404C	2A	5F
553B	20	EA
553C	93	EA
553D	FE	EA
9D5E	wechselt	46
9D5F	wechselt	56

dung auf friedlichem, und uns daher erlaubten Gebieten finden kann, welche über das theoretische Interesse hinaus größte praktische Bedeutung hat, möchten wir auf die Möglichkeiten hinweisen, welche sich für unsere Gerätebauindustrie und Forschungsinstitute durch das Ausziehen einer gegebenenfalls gemeinschaftlichen Entwicklung ergeben. Die deutsche Leistungsfähigkeit auf feinmechanischem Gebiet könnte auf der Grundlage der bereits geleisteten theoretischen Entwicklungsarbeit unserer Gerätebauindustrie eine führende Stellung auf diesem für die Zukunft wichtigen Gebiet schaffen.

Zuse Ingenieurbüro, 1947

Literatur

- [1] Joepgen, Hans-Georg: Vorsicht – Falle! Die „Null-Probleme“ binärer Basic-Interpreter. FUNKSCHAU 1980, Heft 2.
- [2] Joepgen, Hans-Georg: „Kommissar“ deckt Rechenungenauigkeit auf. Hobbycomputer 1. Sonderheft von FUNKSCHAU, ELO und ELEKTRONIK.
- [3] Kaucher, Edgar; Klatte, Rudi; Ullrich, Christian: Höhere Programmiersprache. BI-Wissenschaftsverlag, Mannheim.
- [4] Rint, Curt (Herausgeber): ALGOL-FORTRAN-BASIC-FOCAL – eine vergleichende Darstellung. Dr. Alfred Hüthig Verlag, Heidelberg.
- [5] Bowles, Kenneth L (und andere): Apple Pascal Reference Manual. Apple Computer Incorporated, Cupertino, Kalifornien.
- [6] The 6502 Program Exchange (Herausgeber): „Mini Manual for FCL-65 and FCL-65E“. The 6502 Program Exchange, 2920 Moana, Reno, NV 98509, USA.
- [7] The FOCAL Group, Colorado School of Mines Computing Center (Herausgeber): FOCAL-65. A User's Guide to FOCAL for the 6502 Microprocessor. The 6502 Program Exchange, 2920 Moana, Reno, NV 89509, USA.
- [8] Joepgen, Hans-Georg: Amerika – so weit wie der nächste Briefkasten. Wie kauft man in den Vereinigten Staaten ein? Hobbycomputer 2. Sonderheft der FUNKSCHAU.
- [9] Joepgen, Hans-Georg: Eindeutig, zweideutig – das Gleichheitszeichen in Basic. FUNKSCHAU 1980, Heft 10.
- [10] Joepgen, Hans-Georg: „Compreter“ und „Interpreter“ auf dem Vormarsch. FUNKSCHAU 1980, Heft 14.