

Hans-Georg Joepgen

Focal auf dem Vormarsch

1. Teil

In den USA macht ein neuer Hochsprachen-Interpreter für 6502-Mikrocomputer von sich reden, der über eine Fülle von Vorzügen gegenüber Basic-Übersetzern verfügt. Wichtigster Vorteil, der diese Neuversion von Focal besonders für den Gebrauch an unseren Schulen geeignet macht: Sämtliche Anweisungen an den Rechner können auch in Deutsch erfolgen. Über erste Betriebserfahrungen mit Sprache und Interpreter berichtet der folgende Beitrag.

Focal ist eine Kurzform von „Formulating on-line calculations in algebraic language“ – was man am besten mit „Algebraische Formulierungsvorschrift für die Berechnungen im Konsolenbetrieb“ übersetzt. Die Urform der Sprache entstand in den Sechzigerjahren, als es noch allgemein üblich war, Computern Programme und Ausgangsdaten auf Lochkarten zu übergeben und der „On-Line-Betrieb“ an einer Konsole noch eine Besonderheit war, auf der die Schöpfer des Sprachnamens gesondert hinweisen wollten. Zwar gab es, als Focal entstand, das ebenfalls auf Interaktion per Konsole angelegte Basic bereits, aber diese Sprache wurde von Profis nicht ernstgenommen (wie auch heute gewisse Zeitgenossen, die lediglich Ur-Basic kennen, moderne Basic-Versionen aus Unkenntnis nicht zu den höheren Programmiersprachen rechnen mögen). Die Firma Digital Equipment, treibende Kraft hinter der Focal-Entwicklung, rüstete lange Zeit ihre berühmten Mini-computer mit Focal aus. Von dort her eroberte sich die Sprache viele Anwendungsgebiete; so arbeiten zum Beispiel Datenanlagen zur Sicherung des S- und U-Bahn-Verkehrs in vielen europäischen Städten unter Focal: auch die unlängst in Betrieb genommene Schnellbahnstrecke Stuttgart-Marbach wird von Rechnern kontrolliert, deren Betriebssoftware ursprünglich zu einem beträchtlichen Teil in Focal formuliert wurde.

Bereits 1977 begann ob der bekannten Verbreitung dieser Sprache in vielerlei professionellen Minicomputer-Berei-

chen eine Gruppe amerikanischer 6502-Freunde um den bekannten Fachautor Wayne Wall mit der Arbeit an einem Mikro-Focal-Interpreter, der am 11.4.78 den letzten Test bestand (die Kopie eines Listings von diesem Tage ist das Glanzstück in des Verfassers Sammlung „historischer“ EDV-Dokumente). Den Vertrieb dieser Interpreter in Form von Tonbandkassetten übernahm später die Software-Börse „The 6502 Program Exchange“ in Reno, Nevada. Es wurden zwei Versionen angeboten, FCL-65 und später FCL-65E, die freilich, was Rechengeschwindigkeit angeht, mit den etwa zur gleichen Zeit entwickelten Mi-

crosoft-Basic-Dialekten für Apple und Commodore nicht mithalten konnten. Seit 1980 verkauft „The Program Exchange“ nun für bescheidene 30 Dollar einen wesentlich schnelleren Interpreter – „FOCAL-65“, der hier besprochen wird.

Blockstruktur durch Gruppen-Nummern

Im folgenden kann keine komplette Einführung in das Programmieren mit Focal gegeben werden, es sollen dagegen einige Eigenheiten dieser leistungsfähigen Sprache beispielhaft herausgegriffen werden und typische Eigenschaften beleuchtet werden. Dazu dient das Listing aus Bild 1: Man erkennt, daß jeweils einer Anzahl von Zeilen noch eine weitere Nummer vorangestellt ist, die „Gruppen-Nummer“. Über sie kann man ganze Programm-Moduln gemeinschaftlich behandeln, sie gemeinsam aufrufen oder zusammen löschen. Die Anweisung DURCHFUEHRUNG 40 in Gruppe 20, Zeile 30, bedeutet zum Beispiel, daß alle Zeilen der Gruppe 40 als Unterprogramm aufzurufen sind. Nebenbei: Man

```

_WIEDERGABE
C FOCAL-65 (V3E) 1-0CT-79
10 01 C
10 02 C      *** EINGABE - DEMONSTRATION ***
10 03 C
10 04 C
10 05 C      (HANS-GEORG JOEPGEN, 1 9 1980)
10 06 C
10 07 TEXTFORMAT % 5.00

20 10 SETZE S1=111, S2=222, S3=333, S4=444, S5=555, S6=666
20 20 TV-AUSGABE !, !, " *** 50 SCHOEN SPIELT FOCAL: ", !, "
-----", !, !
20 30 DURCHFUEHRUNG 40
20 40 TV-AUSGABE " *** SUMME AUGENBLICKLICH: ", S, !, !
20 50 ANFRAGE " --- WAS KOMMT DAZU? " S7
20 60 INSOERN (S7), 30 1.
20 70 SETZE S=S+S7; GEHNACH 20. 4

30 10 TV-AUSGABE !, !, "DAS WAER'S ADIEU. -" !
30 20 QUITTIEREN

40 10 SETZE S=S1+S2+S3+S4+S5+S6

```

Bild 1. So sieht ein Focal-Programm in deutscher Langnotation aus

kann auch DO 40 oder, kürzer noch, D 40 schreiben. Der Interpreter wertet nur den ersten Buchstaben von Kommandos aus – so kommt es, daß er die in *Tabelle 1* genannten englischen und deutschen Schlüsselworte gleichermaßen versteht. Zurück zum Programm EINGABE-DEMONSTRATION: Zeile 10.07 setzt das Format aller Zahlenausgaben auf 5 Vorkomma-Stellen und null Nachkommastellen fest. Die Vorkomma-Formatierung ersetzt führende Nullen durch Leer-Stellen; dies ermöglicht sauberen Tabellendruck. Die Vorschrift über die auszugebenden Nachkommastellen rundet und füllt gegebenenfalls mit Nachnullen: Was dies bedeutet, wissen besonders PET-Besitzer und Benutzer der beiden Standard-Basics auf den Maschinen der Apple-Klasse zu schätzen, die durch das Fehlen von derlei Möglichkeiten zu allerlei Klimmzügen gezwungen sind.

Zur Zeile 20.10: Zweideutigkeiten betreffend Gleichheitszeichen wie in Basic [9] gibt's in Focal nicht. In der folgenden Zeile ist das Ausrufezeichen von besonderer Bedeutung: Es bewirkt CR/LF, also den Sprung des Cursors und damit den Wechsel des Druckortes auf dem Schirm zum Anfang der nächsten Bildschirmzeile. Man erkennt weiter, daß die hintereinander zu druckenden Ziffern und Texte im Programm durch jeweils ein Komma getrennt werden, das hier nicht dem Cursorsprung in Tabellierpositionen entspricht, wie das in Basic der Fall ist.

Bis auf Zeile 20.60 erklärt sich das Programm selbst – dort finden wir ein I-Statement (im Langtext INSOFFERN oder IF). Die syntaktische Vorschrift dafür: Nach dem Befehl I in Klammern ein mathematischer Ausdruck oder der Name einer Variablen, danach haben drei Zeilennummern mit vorangesezter Gruppen-Nummer zu stehen, diese durch Komma getrennt. Ist nun der Wert in der Klammer negativ, erfolgt Sprung zur ersten der drei Zeilen, ist der Klammerwert Null, geht's in die zweite Zeilennummer, ist er größer als Null, wird in die dritte angegebene Zeile gesprungen.

Dies erinnert an FORTRAN – freilich, man muß nicht alle drei Angaben hinter der Klammer machen, kann bis zu zwei davon weglassen und setzt nur die betreffenden Trenn-Kommata. In Zeile 20.60 ist die Kleiner-Null-Ab sprungadresse ebenso weggelassen wie die Größer-Null-Sprungzieladresse. Dies bedeutet: Nur wenn der Eingabewert in der Variablen S7 wirklich Null ist, erfolgt Ab sprung nach 30.1, ansonsten geht's

Tabelle 1: Einige Focal-Kommandos

Kurz-Zeichen	Deutsches Schlüsselwort	Englisches Schlüsselwort	Funktionsentsprechung in Basic etwa z. B.
A	ANFRAGE	ASK	INPUT „TEXT“; A\$
C	COMMENTAR	COMMENT	REM
D	DURCHFUEHRUNG	DO	bedingt: GOSUB, DEFFN, RUN
E	ENTFERNE	ERASE	CLEAR: CLR
E A	ENTFERNE ALLES	ERASE ALL	NEW
F	FOLGE	FOR	FOR ... NEXT
G	GESAMTLAUF	GO	RUN
G	GEHNACH	GO	GOTO
I	INSOFFERN	IF	IF, aber mit dreifacher Verzweigung (wie FORTRAN II) wie EDIT in DAI-Basic ähnlich ON, aber Dreifachweiche
M	MODIFIZIERUNG	MODIFY	
O	OPERATOR-ROUTINE	ON	STOP, BREAK, END
Q	QUITTIEREN	QUIT	CMD 0; PK # 0
R	RUECKSTELLEN	RESTORE	RETURN
R	RUECKKEHR	RETURN	RETURN
S	SETZEN	SET	LET
T	TV-AUSGABE	TYPE	PRINT
W	WIEDERGABE	WRITE	LIST

mit Zeile 20.70 und anschließend mit 20.40 weiter (es heißt GEHNACH 20.4, weil Folge-Nullen in Sprunganweisungen weggelassen werden dürfen). Bemerkenswert finden Basic-Programmier-

er schließlich sicherlich noch, daß es in Focal im Normalfall keiner Rücksprunganweisung zum Subroutinen-Abschluß bedarf – Zeile 40.10 enthält kein RETURN.

Dateneingaben auch indirekt möglich

Bis hierhin ist schon erkennbar geworden, daß man in Focal mit wesentlich konzentrierterem Quellencode arbeiten kann, als dies in Basic oder dem noch „geschwätzigeren“ Pascal möglich ist. Nun aber zeigt der Focal-Interpreter von Wayne Wall eine Eigenschaft, die Basic- und Pascal-Freunde sicherlich mit Neid erfüllt: Wir starten das soeben beschriebene Programm auf unserer ITT 2020 (Bild 2) und erleben, daß zu unserer Anfangssumme brav 1 addiert und von ihr sodann 10 subtrahiert wird. Als dritte Eingabe folgt nun jedoch kein Klartext-Wert mehr, sondern der Name der vom Programm zuvor mit dem Wert 111 belegten Variablen S1 – und siehe da, was Interpreter anderer Sprachen üblicherweise, bewältigt unsere 2020 unter FOCAL-65 auf Anhieb, der Variablenwert wird anstandslos weiterverarbeitet! Als nächstes geben wir weder Klartext-Wert noch Variablen-Namen ein, sondern einen mathematischen Ausdruck, einen Bruch: Anstandslos akzeptiert. Nun werden wir ausgesprochen mutig und versuchen, während das Programm läuft, quasi unterwegs eine neue Variable zu definieren, und geben ein X=7.

```

_DURCHFUEHRUNG
+++ SO SCHON SPIELT FOCAL :
-----
+++ SUMME AUGENBLICKLICH: 2331
--- WAS KOMMT DAZU? 1
+++ SUMME AUGENBLICKLICH: 2332
--- WAS KOMMT DAZU? -10
+++ SUMME AUGENBLICKLICH: 2322
--- WAS KOMMT DAZU? S1
+++ SUMME AUGENBLICKLICH: 2433
--- WAS KOMMT DAZU? 5000/2
+++ SUMME AUGENBLICKLICH: 4933
--- WAS KOMMT DAZU? S1+100+53
+++ SUMME AUGENBLICKLICH: 5477
--- WAS KOMMT DAZU? X=7
+++ SUMME AUGENBLICKLICH: 5484
--- WAS KOMMT DAZU? X
+++ SUMME AUGENBLICKLICH: 5491
--- WAS KOMMT DAZU? 0
DAS WAER'S. ADIEU. -
    
```

Bild 2. Statt Klartext-Werten Variablen-Namen oder mathematische Ausdrücke eingeben, auf Wunsch sogar neue Variable generieren, während das Programm schon läuft: Das alles kann FOCAL-65

```

-C AUSDRUCK ALLER VARIABLENWERTE:
-
-T #
S1( 0)= 111
S2( 0)= 222
S3( 0)= 333
S4( 0)= 444
S5( 0)= 555
S6( 0)= 666
S8( 0)= 5491
S7( 0)= 9
X8( 0)= 7
    
```

Bild 3. Hilfreich bei der Fehlersuche: Blitzanalyse aller Variablen

Der Rechner verarbeitet den Summanden 7 brav – und wie die vorletzte Eingabe X zeigt, wird der Name der neuen Variablen fortan akzeptiert. Als Schlußzeichen geben wir zu guter Letzt Null ein – das Programm verabschiedet sich. Für uns besteht damit Gelegenheit, einen weiteren Vorzug von FOCAL-65 vorzuführen (Bild 3): Wir tippen ein T als Abkürzung für „TV-AUSGABE“ oder auch „TYPE“, danach Leertaste (muß hinter jedem Kommando als Begrenzer stehen, wenn kein CR-Zeichen [Return] folgt), und schließlich ein Dollarzeichen. Dieser Spezialbefehl, der übrigens auch im Programm selbst stehen kann, bewirkt die Ausgabe von Namen und Werten aller im Augenblick definierten Variablen! Man beachte, daß auch unser eingeschobenes X aufgeführt wird. An der ausgedruckten Liste fällt weiterhin auf, daß den Variablenamen Klammern folgen, in denen eine Leerstelle und eine Null stehen. Des Rätsels Lösung: Schlichte Variable mit Namen wie S1 zum Beispiel kennt FOCAL-65 in Wahrheit nicht, da geht's ein wenig komplizierter zu.

Datentypen selbstgebaut

Um die indexgesteuerte Verarbeitung großer Datenmengen zu erleichtern, und dies unabhängig davon, ob es hierbei um Texte, numerische Daten in Byte-Verschlüsselung und problemangepaßten Formaten oder um bitcodierte Sachverhalte geht: Dafür gaben die Väter von Urfocal und Wayne Wall ihren Interpretern die Fähigkeit mit auf den Weg, höchst flexibel mit Zeichenketten umzugehen. Zusätzlich wurden Routinen ho-

Tabelle 2: Funktionen in FOCAL-65 (Version 1980; Auszug)

FABS (X)	Absoluter Wert von X	FSTI (X, AS, Y)	Rechnerausgabe erfolgt nicht länger in den spezifizierten Ausgabekanal oder auf den Schirm, sondern in den String AS – solange, bis er mit X Zeichen gefüllt ist oder ein Schlußzeichen mit dem Dezimalcode Y im Datenstrom erscheint
FINT (X)	Wert einer Dezimalzahl ohne Nachkommastellen	FSTO (X, AS, Y)	Kommandos und Daten kommen nicht länger von der Konsole oder aus dem spezifizierten Eingabekanal, sondern aus dem String AS -Syntax wie oben
FINR (X)	Wie oben, jedoch gerundet	FSQT (X)	Quadratwurzel
FRAN (X)	Aufruf des Zufallszahlen-Generators mit verschiedenen Betriebsarten	FLN (X)	Natürlicher Logarithmus
FIDV (X)	Künftige Eingaben nicht länger von Konsole, sondern über den Datenkanal mit der Nummer X (Floppy, Band u. a.)	FLG (X)	Zehner Logarithmus
FODV (X)	Ausgabe nicht länger auf den Schirm, sondern in den Datenkanal mit der Nummer X	FXPN (X)	Exponential-Funktion
FCHR (X)	Dezimalcode des ASCII Zeichens im String wird gerufen	FCAL (X, Y, Z)	Monitorroutine auf Seite x, Adresse Y wird mit dem Inhalt von Z im Akku aufgerufen
FOUT (X)	Das ASCII Zeichen mit dem Dezimalcode X wird ausgegeben		(Trigonometrische Funktionen, Kegelschnitt-Funktionen, Statistik-Funktionen, Ausgabe in Exponentenform oder im sogenannten „Ingenieurformat“ und Naturkonstanten werden bei Bedarf der UPRO-Bibliothek entnommen und am Schluß des Programmtextes angefügt. Hierzu dient die Gruppe 99)
FECH (X)	Die Wiederholung einlaufender Zeichen auf dem Schirm wird eingeschaltet („Echoing“)		Für die Grundrechenarten stehen die üblichen Operatorzeichen zur Verfügung. Exponenten werden mit einem hochgestellten Haken gekennzeichnet.
FMEM (X, Y)	Inhalt der Speicherzelle Y auf Seite X wird gerufen		
FMEM (X, Y, Z)	Wie oben, zusätzlich Transfer von Z in die Zelle nach Auslesen		
FSBR (X, Y)	Die Subroutine in Gruppe X wird gerufen. Parameterübergabe in Y. Parameter-Rückgabe im Array &0..&9		
FISL (A, AS)	Ein String AS mit Raum für A Zeichen wird geschaffen		

hen Freiheitsgrades zur Manipulation von Feldern vorgesehen. Da nun der ganze Interpreter möglichst schlank geraten sollte, denn Adreßraum auf Mikrocomputern ist knapp und FOCAL-65 belegt bescheidene 8 KByte exklusiv Benutzer-Programm und Page-Zero-Zellen, fiel bei derlei Sparmaßnahmen die nichtindizierte Variable dem Rotstift zum Opfer. Ruft man dergleichen Variable nun auf, wie wir es oben getan haben, so stellt der Interpreter stattdessen eine indizierte Variable mit Index Null zur Verfügung. Mit anderen Worten: S1 wird als abgekürzte Schreib-

weise für Elemente eines eindimensionalen Datenfeldes und dort des Gliedes mit dem Index 0 aufgefaßt: S1 (0) also. Nach diesem Ausflug in Geschichte und Struktur von FOCAL-65 folgt im nächsten Heft der Bericht über Betriebserfahrungen mit FOCAL auf einem ITT-2020, eine kritische Betrachtung zu Rechenzeit und Rechengenauigkeit sowie die Vorstellung eines Focal-Programms aus Spezialisten-Feder, des weitgerühmten mathematischen Konversionsprogramms „BASE“ von Larry Fish.

(Fortsetzung folgt)

Hans-Georg Joepgen

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 („INSOFERN“ 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.

Augenfälliger 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 Winkelnzüge 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 Palsoft 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

```

_JH
C FOCAL-65 (V3E) 1-9CT-79
20 01 A "TESTZAHL? ",A
20 02 O (A-100)21,22,23
20 03 T " HUNDERT -",1;G 20.01
21 01 T "KLEINER ALS "
22 01 T "GLEICH GROSS WIE "
23 01 T "GROSSER ALS "

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

```

Bild 4. Eine Mini-Aufgabe in Focal

```

JLIST
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

JRUN

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 Oktalzahl 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 A\$ und B\$ 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 entfaltet, 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; RO („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 erprobten. 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 Kasette 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 eines 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 wiederum 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 Palssoft gerechnet. Zu den Schnellsten im Lande zählt der neue Interpreter damit durchaus nicht. Als ungeschön empfindet der Verfasser weiter, daß beim Interpreter zwar erfolgreich mit jedem Byte gezeitet wurde, daß der Programmtext selbst sich aber verdichtet 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.

```

_MH
C FOCAL-65 (V3E) 1-0CT-79
10.10 C
10.20 C  FALLTEST
10.30 C  -----
10.40 C
20.10 F N=-1. 1.1:0 30
20.20 0
30.10 T % 2.05.N " ", % 2.05: " ,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 „Falltest“ freilich versagt auch Focal

FOCAL-65 – für wen?

Die weiten Freiräume, die der neue Interpret 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 ausgeschriebenem, 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 Programmiersprache. 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.

Schlußwort: 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 Rechengenäte ihre umfassendste Anwen-

```
C FOCAL-65 <V3E> 1-8CT-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 FISL(16, A#, 16, B#): A "BASE OF FIRST NUMBER: ", B1
1 20 S D1=FSTR(3, B1): T "FIRST NUMBER: "; S J=FSTI(D1, A#, 13): T !
1 25 A "BASE OF SECOND NUMBER: ", B2: S D2=FSTR(3, B2)
1 27 T "SECOND NUMBER: "; S J1=FSTI(D2, B#, 13): T !
1 30 T "OPERATION: "; S O=FCHR(C): 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: D 2 1: S B=Z
1 50 S FODV(C#): T "S C=A": S FOUT(O): T "B", !, R: O: 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=#(X)
2 31 S Y=#(X)
3 10 I (8-4), , 3 5: D (3.2+8/100): R
3 22 S 8=16
3 23 S 8=11
3 24 S 8=9
3 50 I (8-15), , 3 9: I (8-9), , 3 8: I (8-6): 3 6, 3 6, 3 7
3 60 S 8=7: R
3 70 S 8=6: R
3 80 S 8=5: R
3 90 S 8=4: R
4 10 I (B3-10), 4 35: 0 (C)4 5: S D=FSTR(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 !: G
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

ding 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; Ulrich, 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.