

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 X 5 00

20 10 SETZE S1=111, S2=222, S3=333, S4=444, S5=555, S6=666
20 20 TV-AUSGABE '!,', " *** SO 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; GERNACH 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 Interpretierer 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 INSOFERN 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-Absprungadresse ebenso weggelassen wie die Größer-Null-Sprungzieladresse. Dies bedeutet: Nur wenn der Eingabewert in der Variablen S7 wirklich Null ist, erfolgt Absprung 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	INSOFERN	IF	IF, aber mit dreifacher Verzweigung (wie FORTRAN II) wie EDIT in DAI-Basic ähnlich ON, aber Dreifachweiche
M	MODIFIZIERUNG	MODIFY	STOP, BREAK, END
O	OPERATOR-ROUTINE	ON	CMD 0; PR # 0
Q	QUITTIEREN	QUIT	RETURN
R	RUECKSTELLEN	RESTORE	LET
R	RUECKKEHR	RETURN	PRINT
S	SETZEN	SET	LIST
T	TV-AUSGABE	TYPE	
W	WIEDERGABE	WRITE	

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-Programmierer 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-Interpretierer 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 Interpretierer anderer Sprachen übelnehmen, 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+S3
+++ 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
S0( 0)= 5491
S7( 0)= 0
S8( 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-Verschüsselung 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)**

<b>FABS (X)</b>	Absoluter Wert von X	<b>FSTI (X, AS, Y)</b>	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
<b>FINT (X)</b>	Wert einer Dezimalzahl ohne Nachkommastellen	<b>FSTO (X, AS, Y)</b>	Kommandos und Daten kommen nicht länger von der Konsole oder aus dem spezifizierten Eingabekanal, sondern aus dem String AS-Syntax wie oben
<b>FINR (X)</b>	Wie oben, jedoch gerundet	<b>FSQT (X)</b>	Quadratwurzel
<b>FRAN (X)</b>	Aufruf des Zufallszahlen-Generators mit verschiedenen Betriebsarten	<b>FLN (X)</b>	Natürlicher Logarithmus
<b>FIDV (X)</b>	Künftige Eingaben nicht länger von Konsole, sondern über den Datenkanal mit der Nummer X (Floppy, Band u. a.)	<b>FLG (X)</b>	Zehner Logarithmus
<b>FODV (X)</b>	Ausgabe nicht länger auf den Schirm, sondern in den Datenkanal mit der Nummer X	<b>FXPN (X)</b>	Exponential-Funktion
<b>FCHR (X)</b>	Dezimalcode des ASCII-Zeichens im String wird gerufen	<b>FCAL (X, Y, Z)</b>	Monitorroutine auf Seite x, Adresse Y wird mit dem Inhalt von Z im Akku aufgerufen
<b>FOUT (X)</b>	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)
<b>FECH (X)</b>	Die Wiederholung einlaufender Zeichen auf dem Schirm wird ein- oder ausgeschaltet („Echoing“)		Für die Grundrechenarten stehen die üblichen Operatorzeichen zur Verfügung. Exponenten werden mit einem hochgestellten Haken gekennzeichnet
<b>FMEM (X, Y)</b>	Inhalt der Speicherzelle Y auf Seite X wird gerufen		
<b>FMEM (X, Y, Z)</b>	Wie oben, zusätzlich Transfer von Z in die Zelle nach Auslesen		
<b>FSBR (X, Y)</b>	Die Subroutine in Gruppe X wird gerufen, Parameterübergabe in Y, Parameter-Rückgabe im Array &0..&9		
<b>FISL (A, AS)</b>	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)