

RUHR-UNIVERSITÄT BOCHUM

Arbeitsberichte
des
Rechenzentrums

Direktor: o.Prof. Dr. Hartmut Ehlich

Nr. 7708

ISSN 0341-0358

ERZEUGE

eine Erweiterung der Kommandosprache

von

H. Wupper

Bochum im Juni 1977

1. Auflage 1977 Copyright by Rechenzentrum der Ruhr-Universität

Vervielfältigung oder Nachdruck, auch auszugsweise, nur unter
Quellenangabe bei Überlassung von 3 Belegexemplaren gestattet.

ISSN 0341-0358

Rechenzentrum der Ruhr-Universität Bochum
Universitätsstraße 150, Gebäude NA
Postfach 102148

4630 Bochum 1

INHALTSVERZEICHNIS

	Seite
1. VORÜBERLEGUNG	5
2. ZIELSETZUNG	7
3. BESCHREIBUNG DES ERZEUGF-KOMMANDOS	8
3.1. Kommandoblätter	9
3.2. Einführung	25
3.3. Wie entsteht ein Ergebnis?	28
3.4. Wie werden die Teilwerte ausgewählt?	31
3.5. Was geschieht mit dem Ergebnis?	32
3.6. Wann entstehen mehrere Ergebnisse?	33
3.7. Protokollierung	36
4. SCHNITTSTELLEN ZUM PROGRAMMIERSYSTEM	37
4.1. BEREICH	37
4.2. Formeln	38



1. VORÜBERLEGUNG

Im TNS440 ist Kommandosprache ein Oberbegriff für mindestens zwei verschiedene formale Sprachen; Gegenstand dieser Überlegungen ist jedoch nur die Sprache der Programmiersystem-Kommandos.

Diese Kommandosprache ist außergewöhnlich flexibel und leistungsfähig konzipiert, dabei aber leicht erlernbar. Leider bestehen jedoch drei Nachteile:

1. Trotz des guten Sprachkonzeptes sind einzelne Kommandos auf ausgesprochen unglückliche Art implementiert worden:

Einerseits wurde bei der Definition der Bedeutung und erlaubten Besetzung von Spezifikationen in den seltensten Fällen auf Verwandtschaft mit Spezifikationen anderer (im Falle BIBVERLAGERE sogar desselben) Kommandos Rücksicht genommen, andererseits wird oft bei ein und derselben Spezifikation eine Reihe von Angaben verlangt, die man in der Praxis oft einzeln besetzen möchte. Diese beiden Dinge tragen dazu bei, daß manchmal die durch das Prozedurkonzept gegebenen Möglichkeiten nicht ausgenutzt werden können. (Beispiele finden sich in den Arbeitsberichten 7209 und 7403.)

2. Einige wichtige Elemente von Programmiersprachen, z. B. die Arithmetik, fehlen in der Kommandosprache. Auf den ersten Blick erscheint es fraglich, ob die Kommandosprache überhaupt so etwas benötigt; schon allein aus Laufzeitgründen schreibt man ja in ihr keine größeren Programme. In der Praxis zeigte sich jedoch, daß von Benutzern oft - z. B. im Zusammenhang mit der Verwaltung von Dateien und Datenträgern - Fortran-Programme geschrieben wurden, die nach kurzen Abfragen oder Berechnungen ihrerseits wieder mit KOMMDO Kommandos ausführen lassen. In vielen Fällen, z. B., wenn nur täglich eine Dateifolgernummer um 1 erhöht werden soll, wird so wahrscheinlich mehr Zeit verbraucht, als ein geeignetes Kommando benötigen würde - vom Platz, den alle diese kleinen Programme benötigen, ganz zu schweigen.

3. Die Variablen der Kommandosprache, nämlich die internen Namen, verhalten sich nicht wie die Variablen gewohnter Programmiersprachen (z. B. ist ein Wertetausch zweier interner Namen nicht möglich), weil bei Zuweisungen nur Verweise eingetragen, nicht Werte kopiert werden. Schleifen sind nicht, Sprünge nur eingeschränkt möglich; an Rekursivität ist überhaupt nicht zu denken.

Der erstgenannte Punkt bemängelt Schwächen der einzelnen vorhandenen Kommandos, die mit dem Konzept der Sprache nichts und mit der Konstruktion des Entschlüsslers wenig zu tun haben. Daß die im Handbuch beschriebenen Kommandos grundlegend abgeändert werden, ist nicht zu erwarten; Abhilfe wäre also nur durch geeignete "Einrahmung" oder "Verkleidung" zu schaffen.

Der zweite Punkt beklagt das Fehlen von Leistungen, die sich durch geeignete Kommandos ohne weiteres erbringen ließen. Ein Beispiel hierfür ist das Bochumer Kommando ZUSTAND, das auf Kommandoebene viele nützliche Abfragen und Prüfungen sowie die Manipulation von Wahlschaltern erlaubt.

Die im dritten Punkt genannten Nachteile liegen im Konzept der Sprache bzw. in der Konstruktion des Entschlüsslers begründet; im folgenden soll jedoch gezeigt werden, daß auch hier ohne Eingriff in das Wesen der Kommandosprache eine Verbesserung der Lage möglich ist.

2. ZIELSETZUNG

Abhilfe ließe sich schaffen durch einen "Kommandowandler", einen Operator mit vielen Eingängen und vielen zugehörigen Kommandos. Ähnlich, wie PS&KINFORM das LFINFORMIERE-Kommando auf INFORMIERE abbildet - das läßt sich nicht durch eine Kommandoprozedur erreichen -, könnte dieses Programm beliebigen Ansprüchen genügende neue Kommandos auf die unschönen vorhandenen abbilden.

Außerdem müßten einige ganz neue Leistungen wie SUMME (oder LET wie in Basic?), IF, DO o. ä. geschaffen werden.

Dabei gerät man allerdings, wie die Erfahrung bei entsprechenden Versuchen in Bochum zeigte, sehr schnell ins Uferlose.

Besser als viele einzelne Kommandos oder ein Kommando mit vielen "Tätigkeiten" (vgl. LFVERWALTE) oder "Modi" - ist die Bereitstellung einiger weniger grundlegender Verfahren, die beliebig kombiniert werden dürfen, in einen "Metakommando".

ERZEUGE versucht dies; die wichtigsten der "orthogonal kombinierbaren" Verfahren sind

- Verkettung mehrerer Spezifikationswerte zu einem String,
- Berechnung arithmetischer Ausdrücke,
- Verarbeitung von Teilwertlisten,
- rekursive und iterative Ausführung von erzeugten Zeichenfolgen als Kommandos; dabei kann das Fluchtsymbol durch ein beliebig vorzuzugendes Zeichen dargestellt werden.

3. BESCHREIBUNG DES ERZEUGE-KOMMANDOS

3.1. Die folgenden Handbuchblätter enthalten eine formale Beschreibung des Kommandos.
Zum Einstieg dürfte es sich allerdings empfehlen, zuerst die Abschnitte 3.2. bis
3.5. zu lesen und bei Bedarf Vergleiche mit 3.1. anzustellen.

Einige mehr für Rechenzentren als für Benutzer wichtige Leistungen sind auf den
Handbuchblättern nicht erwähnt.

ERZEUGE

ERZEUGE

Manipulationen mit Elementen der Kommandosprache

Spezifikation:

- 1 ZIEL : Gewünschte Dienstleistung oder zu besetzender interner Name
-
- 2 QUELLE : Ausgangstext
- 3 TEILWERTE : Texte zur Modifikation der QUELLE
- 4 KRITERIUM : Auswahl- oder Abbruchkriterien
- 5 MODUS : Angabe der gewünschten Interpretation der Spezifikationen TEILWERTE und KRITERIUM
- 6 MAL : Auszeichnung eines Zeichens als Mal
- 7 PROTOKOLL : Steuerung der Protokollierung

Kommando für das Programmiersystem

Einschränkung:

Wirkung:

Elemente der Kommandosprache - Normal- und Fremdstrings - können auf vielfältige Art verarbeitet werden.

Möglich sind z. B.

- Verkettung mehrerer Spezifikationswerte zu einer Zeichenfolge,
- Auswahl bestimmter Teilwerte aus einer Liste,
- Interpretation von (Teil-)Zeichenfolgen als Formeln bzw. arithmetische Anweisungen,
- Bestimmung der Anzahl von Teilwerten einer Liste.

Die Ergebnisse werden wie unter ZIEL angegeben einem internen Namen zur Weiterverarbeitung zugewiesen oder sie bewirken eine Dienstleistung (Ausführung oder Definition von Kommandos).

Die Bearbeitung geht von dem unter QUELLE angegebenen Text aus, der je nach MODUS mit Teilwerten von TEILWERTE modifiziert wird. In Abhängigkeit von MODUS werden die Angaben unter KRITERIUM als Indizes oder Abbruchkriterien aufgefaßt.

Das als MAL definierte Zeichen gestattet besondere Bearbeitung beliebiger Stellen des Textes.

ERZEUGE

formal:

$\langle \text{ERZEUGE-Kommando} \rangle ::= \diamond \text{ERZEUGE} [, [\langle \text{Spezifikationsname} \rangle =] \langle \text{Spezifikationswert} \rangle]^\infty$

$\langle \text{Spezifikationsname} \rangle ::= \text{ZIEL} \mid \text{QUELLE} \mid \text{TEILWERTE} \mid \text{KRITERIUM} \mid \text{MÖDUS} \mid \text{MAL} \mid \text{PRÖTÖKÖLL}$

Beispiel:

```
#TDEKL.,C #TEINTR.,A,INF.=/  
DREI CHINESEN MIT DEM KONTRABASS  
SASSEN AUF DER STRASSE UND ERZAEHLTEN SICH WAS.  
DA KAM DIE POLIZEI UND FRAGT: "WAS IST DENN DAS?  
DREI CHINESEN MIT DEM KONTRABASS!"
```

```
#ERZEUGE, KOMMANDO, MODUS=SCHEIFE, TEILWERTE=A^E^I^O^U, MAL=$, QU.=/  
$CODEWANDLE, C, A, 1, BLOCKUNG=1, CODE=/  
ART=2, A=$!1, E=$!1, I=$!1, O=$!1, U=$!1, TREZE1='AF', '30', '0 $DAU.,C,KO^ON#.  
KREIERT: C(0001.00)
```

DIE ZIELDATEI A(0001.00) WURDE ERSTELLT

ENDE TEINTRAGE (7.01) 0.17

ENDE BD&CODE2 (1.02) 0.16

```
DRAA CHANASAN MAT DAM KANTRABASS  
SASSAN AAF DAR STRASSA AND ARZAAHLTAN SACH WAS.  
DA KAM DAA PALAZAA AND FRAGT: "WAS AST DENN DAS?  
DRAA CHANASAN MAT DAM KANTRABASS!"
```

ENDE BD&CODE2 (1.02) 0.15

```
DREE CHENESEN MET DEM KENTREBESS  
SESSEN EEF DER STRESSE END ERZEEHLTEN SECH WES.  
DE KEM DEE PELEZEE END FREGT: "WES EST DENN DES?  
DREE CHENESEN MET DEM KENTREBESS!"
```

ENDE DAUSGABE (14.06) 0.25

ENDE BD&CODE2 (1.02) 0.13

```
DRII CHINISIN MIT DIM KINTRIBISS  
SISSIN IIF DIR STRISSI IND IRZIIHLTIN SICH WIS.  
DI KIM DII PILIZII IND FRIGT: "WIS IST DINN DIS?  
DRII CHINISIN MIT DIM KINTRIBISS!"
```

ENDE DAUSGABE (14.06) 0.16

ENDE BD&CODE2 (1.02) 0.14

```
DROO CHONOSON MOT DOM KONTRABASS  
SOSSON OOF DOR STROSSO UND DRZOOHLTON SOCH WOS.
```

ERZEUGE

ZIEL

ZIEL

Angabe von internen Namen oder Tätigkeiten 1

Spez.-Wert:	TEXT	:	Das Ergebnis wird ausgedruckt
	n	:	Das Ergebnis wird dem internen Namen *n zugewiesen.
	KOMMANDO	:	Das Ergebnis ist eine Kommandofolge, die ausgeführt wird.
	BEREICH	:	Das Ergebnis wird als die Bezeichnung eines Problemkomplexes aufgefaßt, zu dem Kommandos definiert oder ausgeführt werden sollen.

Mehrere Angaben sind durch Apostroph zu trennen.

Obligate Spezifikation z. Kdo. ERZEUGE

anlagenspezifische
Voreinstellung: "undefiniert"

Einschränkung:

Die im Falle BEREICH zulässigen Problemkomplexe sind rechenzentrumsspezifisch.

Wirkung:

In den Modi INDIZIERUNG und SCHLEIFE entstehen wie unter MODUS und QUELLE beschrieben ein oder mehrere Ergebnisse, die der Reihe nach den Teilwerten von Ziel zugeordnet werden. Lautet der zugeordnete Teilwert TEXT, so wird das Ergebnis ausgedruckt; ist er eine natürliche Zahl, so wird das Ergebnis dem dadurch bezeichneten internen Namen zugewiesen. Im Falle KOMMANDO wird das Ergebnis als Kommandofolge interpretiert, im Falle Bereich muß es die Bezeichnung eines zu aktivierenden Problemkomplexes sein.

Entstehen mehr Ergebnisse, als Ziele angegeben wurden, so wird das letzte definierte Ziel allen weiteren Ergebnissen zugeordnet.

Im Modus ANZAHL sind nur interne Namen als Ziele zulässig.

ERZEUGE/ZIEL

formal:

$\langle \text{Wertzuweisung ZIEL} \rangle ::= [\text{ZIEL=}] \langle \text{Zielangabe} \rangle [' \langle \text{Zielangabe} \rangle]^\infty$

$\langle \text{Zielangabe} \rangle ::= \langle \text{Adresse eines internen Namens} \rangle \mid \text{KOMMANDØ} \mid$
 $\text{TEXT} \mid \text{BEREICH}$

$\langle \text{Adresse eines internen Namens} \rangle ::= \langle \text{Ziffer} \neq 0 \rangle [\langle \text{Ziffer} \rangle]^5$

Die Angaben KOMMANDØ und BEREICH dürfen wie Tätigkeitsnamen abgekürzt werden.

Beispiel:

◇ERZ., ZIEL=KOMM. 'BER. '13, , LFI. 'ABK. 'AFFE, MØD.=SCHL.

Bewirkt wird

1. Ausführung des Kommandos ◇LFI. ,
2. Definition der Kommandos aus dem Bereich "Abkürzungen" ,
3. Zuweisung *13=AFFE

ERZEUGE

QUELLE

QUELLE

Angabe des Ausgangstextes zur Verarbeitung

2

Spez.-Wert:

Zulässig ist eine beliebige Folge von Normal- oder Fremdstrings, die durch Apostroph voneinander getrennt werden.

"undefiniert" : Der Ausgangstext enthält keine Zeichen.

Optionale Spez. zum Kommando ERZEUGE

anlagenspezifische

Voreinstellung: "undefiniert"

Einschränkung:

Wirkung:

Alle Teilwerte werden in der angegebenen Reihenfolge (ohne die trennenden Apostrophe und ohne Fremdstringbegrenzer) zu einem Text verkettet.
Durch `<Mal>!<Zeichen>` oder `<Zeichen><Mal>;` wird ein Platzhalter definiert, der je nach MODUS und evtl. KRITERIUM durch einen Teilwert von TEILWERTE ersetzt wird. Die Quelle kann mehrere Platzhalter enthalten, die, wenn die `<Zeichen>` verschieden sind, durch verschiedene, sonst durch gleiche Teilwerte ersetzt werden.

Alle Male, denen kein ! oder ; folgt, werden wie unter MAL beschrieben interpretiert.

Enthält die Quelle keine Platzhalter, so wird einer als am Ende stehend angenommen.

ERZEUGE/QUELLE

formal:

$\langle \text{Wertzuweisung QUELLE} \rangle ::= [\text{QUELLE=}] \left\{ \begin{array}{l} \langle \text{Spezifikationswert} \rangle \\ - \quad [' \langle \text{Spezifikationswert} \rangle]^{\infty} \end{array} \right\}$

$\langle \text{Spezifikationswert} \rangle ::= \langle \text{Normalstring} \rangle \mid / \langle \text{Fremdstring} \rangle \diamond /$

Beispiel:

Q.=/MB($\diamond / '900013' /$) $\diamond /$, ...

ERZEUGE

TEILWERTE

3

TEILWERTE

Texte zur Modifikation der QUELLE

Spez.-Wert:

Zulässig ist eine beliebige Folge von Normal- oder Fremdstrings, die durch Apostroph voneinander getrennt werden.

Einzelne leere Teilwerte müssen als leere Fremdstrings dargestellt werden, da "undefiniert" als Teilwert verboten ist.

"undefiniert" : Die Liste der Teilwerte ist leer.

Optionale Spez. zum Kommando ERZEUGE

anlagenspezifische
Voreinstellung: "undefiniert"

Einschränkung:

Wirkung:

Im Gegensatz zur QUELLE werden die TEILWERTE nicht verkettet; sie bilden vielmehr eine (eindimensionale) Liste, aus der einzelne Elemente ausgewählt werden.

Enthält die Liste zu wenig Elemente, so wird an Stelle eines nicht vorhandenen der letzte Teilwert genommen; TEILWERTE="undefiniert" wird wie TEILWERTE="leerer String" behandelt.

ERZEUGE/TEILWERTE

formal:

$$\langle \text{Wertzuweisung TEILWERTE} \rangle ::= [\text{TEILWERTE} =] \left\{ \begin{array}{l} \langle \text{Spezifikationswert} \rangle \\ [' \langle \text{Spezifikationswert} \rangle]^{\infty} \\ - \end{array} \right.$$
$$\langle \text{Spezifikationswert} \rangle ::= \langle \text{Normalstring} \rangle \mid / \langle \text{Fremdstring} \rangle \diamond /$$

Beispiel:

◇ ERZ., KØMM., /EINSCHL., KAT.◇/, TEILW.=A'B'C, MØD.=SCHL.

Es werden nacheinander die Dateien

KAT.A

KAT.B

KAT.C

eingeschleust.

ERZEUGE

KRITERIUM



KRITERIUM

Auswahl- oder Abbruchkriterien

Spez.-Wert:

n : Index im Falle INDIZIERUNG,
Wiederholungsfaktor im Falle SCHLEIFE

ERFOLG } : Abbruchkriterien für SCHLEIFE
FEHLER }

"undefiniert" bedeutet soviel wie 1 .

Mehrere Angaben sind durch Apostroph zu trennen.

Optionale Spezifikation z. Kdo. ERZEUGE

anlagenspezifische
Voreinstellung: "undefiniert"

Einschränkung:

Wirkung:

Die Bedeutung richtet sich nach dem MODUS:

MODUS=INDIZIERUNG:

Als Kriterien sind nur natürliche Zahlen zulässig, die als Indizes eine Auswahl von Teilwerten der Spezifikation TEILWERTE bewirken.

MODUS=SCHLEIFE:

Abhängig von der Anzahl der Platzhalter und TEILWERTE entstehen mehrere Ergebnisse, denen der Reihe nach je ein Teilwert von KRITERIUM und einer von ZIEL zugeordnet werden.

Bezeichnet der betreffende Teilwert von ZIEL einen internen Namen, so ist die entsprechende Angabe zu KRITERIUM bedeutungslos; in den Fällen KOMMANDO und BEREICH steuert sie die Anzahl der Ausführungen:

n : Die Ausführung erfolgt n mal.

ERFOLG : Die Ausführung wird so lange wiederholt, bis sie erfolgreich gelingt.

FEHLER : Die Ausführung wird solange wiederholt, wie sie fehlerfrei gelingt.

MODUS=ANZAHL:

Die Angabe zu KRITERIUM ist bedeutungslos.

ERZEUGE/KRITERIUM

formal:

$$\langle \text{Wertzuweisung KRITERIUM} \rangle ::= [\text{KRITERIUM} =] \left\{ \begin{array}{l} \langle \text{Kriterium} \rangle \\ [\langle \text{Kriterium} \rangle]^\infty \\ - \end{array} \right\}$$
$$\langle \text{Kriterium} \rangle ::= \langle \text{Zahl} > 0 \rangle \mid \text{FEHLER} \mid \text{ERFØLG}$$

Die Angaben FEHLER und ERFØLG dürfen wie Tätigkeitsnamen abgekürzt werden.

Beispiel:

...,MØD.=SCHL.,KRIT.=1'FEHLER'ERF.'2

ERZEUGE

MODUS

5

MODUS

Art der Interpretation von TEILW. und KRIT.

Spez.-Wert:

- ANZAHL : Die Anzahl der TEILWERTE wird ermittelt.
- INDIZIERUNG : Durch KRITERIUM werden einzelne TEILWERTE zur Modifikation der QUELLE ausgewählt.
- SCHLEIFE : Die QUELLE wird der Reihe nach durch alle TEILWERTE modifiziert; KRITERIUM beeinflusst die Bearbeitung jedes Ergebnisses.

"undefiniert" bedeutet soviel wie INDIZIERUNG .

Optionale Spezifikation z. Kdo. ERZEUGE

anlagenspezifische

Voreinstellung: "undefiniert"

Einschränkung:

Wirkung:

ANZAHL:

Die Anzahl der Teilwerte von TEILWERTE wird allen unter ZIEL angegebenen internen Namen zugewiesen. Von den restlichen Spezifikationen wird lediglich PROTOKOLL ausgewertet.

INDIZIERUNG:

Ist p die Anzahl der (verschiedenen) Platzhalter von QUELLE, so werden die ersten p Teilwerte von KRITERIUM als Indizes aufgefaßt, die die einzusetzenden p TEILWERTE auswählen. Das Ergebnis wird gemäß der ersten ZIEL-Angabe weiterverarbeitet. Wurden bei ZIEL mehr als eine oder bei KRITERIUM mehr als p Angaben gemacht, wird solange analog weiterverfahren, bis alle Angaben ausgewertet sind; dabei wird ggf. von der zuerst erschöpften Spezifikation der letzte Teilwert mehrfach genommen.

SCHLEIFE:

Sei p die Anzahl der Platzhalter und k das Maximum der Anzahlen der Ziele und Kriterien, dann werden in einer Schleife ($1 \leq i \leq k$) die TEILWERTE der Position i bis $i+(p-1)$, also immer p TEILWERTE der Reihe nach, für die Platzhalter eingesetzt. Das Ergebnis wird dem i -ten ZIEL zugeordnet; das i -te KRITERIUM beeinflusst die Ausführung. Ist die Anzahl der Ziele und Kriterien verschieden, so wird von der zuerst erschöpften Spezifikation der letzte Teilwert mehrfach genommen.

ERZEUGE/MØDUS

formal:

$$\langle \text{Wertzuweisung MØDUS} \rangle ::= [\text{MØDUS=}] \left\{ \begin{array}{l} \text{INDIZIERUNG} \\ \text{SCHLEIFE} \\ \text{ANZAHL} \end{array} \right\}$$

Die Angaben dürfen wie Tätigkeitsnamen abgekürzt werden.

Beispiel:

◇ERZ.,13,,A'B'C,MØD.=ANZ.

Die Anzahl der Teilwerte (hier: 3) wird dem internen Namen *13 zugewiesen.

◇ERZ.,KØMM.,,A'B,KRIT.*2,MØD.=IND.

Ist der Wert von *2 gleich 1, wird das Kommando ◇A ausgeführt, ansonsten das Kommando ◇B.

ERZEUGE

MAL

6

MAL

Definition eines Zeichens als Mal

Spez.-Wert:

- z : Angabe genau eines Zeichens
- /z
/z0/ : Im Normalstring verbotene Zeichen können
als Fremdstring angegeben werden.
- n : Angabe des Zentralcodewertes des gewünschten Zeichens
($n \geq 16$)

"undefiniert" bedeutet soviel wie 53 (Zeichen FL)

Optionale Spez. zum Kommando ERZEUGE

anlagenspezifische
Voreinstellung: "undefiniert"

Einschränkung:

Wirkung:

Wird im Text ein Mal gefunden, so richtet sich der Verlauf der weiteren Interpretation nach dem darauffolgenden Zeichen. Im folgenden sei das Mal durch § dargestellt.

§! und §! wirken als Stringklammern. Innerhalb eines so eingeklammerten Strings werden Male nur erkannt, wenn sie weitere, geschachtelte Stringklammern einleiten. Das äußerste Klammernpaar wird entfernt. Auf diese Weise kann das Ergebnis Male enthalten, die erst bei einer nachfolgenden Verarbeitung wirksam werden.

§+ und §+ wirken als Formelklammern. Die Zeichenfolge §+<Formel>§+ wird durch das Ergebnis der Interpretation der <Formel> ersetzt. Eine Formel kann keine weiteren Formelklammern enthalten.

Gleichbedeutend mit §!<Zeichen> ist <Zeichen>§; .

§! definiert einen Platzhalter der Form §!<Zeichen> . Mit gleichem <Zeichen> benannte Platzhalter werden durch gleiche TEILWERTE ersetzt. Die Auswahl der TEILWERTE richtet sich nach dem MODUS und evtl. KRITERIUM.

Folgt dem Mal eine Ziffer, so wird eine genau dreistellige Zahl n erwartet ($0 \leq n \leq 255$). §n wird durch das Zeichen mit dem Zentralcodewert n ersetzt.

In allen anderen Fällen wird das Mal bei einer nachfolgenden Ausführung des Ergebnisses als Kommandofolge zum Fluchtsymbol.

ERZEUGE/MAL

formal:

$\langle \text{Wertzuweisung MAL} \rangle ::= [\text{MAL=}] \langle \text{Zeichen} \rangle$

$\langle \text{Zeichen} \neq * | = | / | , | (|) | ' \rangle$

$\langle \text{Zeichen} \rangle ::= \langle \text{beliebiges Zeichen} \rangle [\emptyset /]$

$\langle \text{natürliche Zahl } n, 16 \leq n \leq 255 \rangle$

Beispiel:

...,MAL= /)

ERZEUGE PROTOKOLL

7

PROTOKOLL

Steuerung der Protokollierung

Spez.-wert: "undefiniert" : Abgesehen von einer kurzen Meldung der MV-Nummer des Operators und der benötigten Rechenzeit werden nur Fehlermeldungen und Warnungen ausgegeben.

&Z : Unterdrückung der Zeitmeldung

A : Ausführliches Protokoll

KØ : Zusätzliche Protokollierung auf Konsole

KW : Keine Ausgabe von Warnungen

(Ø : Objektprotokoll, nur in Fehler- und Testfällen von Interesse)

Mehrere Angaben sind durch Apostroph zu trennen.

Optionale Spez. zum Kommando ERZEUGE

anlagenspezifische
Voreinstellung: "undefiniert"

Einschränkung:

Wirkung:

Durch die Angaben wird der Grad der Protokollierung beeinflusst:

Soll sich ERZEUGE völlig stumm verhalten (abgesehen von Warnungen und Fehlermeldungen), so ist PROTOKOLL=&Z anzugeben. Ein ausführliches Protokoll wird dagegen durch Angabe von A angefordert.

Die Ausgabe von Warnungen läßt sich durch KW unterdrücken.

Im Normalfall werden im Gespräch nur Fehlermeldungen und Warnungen auf Konsole ausgegeben; wird ein ausführliches Protokoll auch auf Konsole gewünscht, ist zusätzlich KØ anzugeben.

Eine Angabe von Ø ist nur sinnvoll, wenn das Protokoll zwecks Fehlerverfolgung weitergereicht werden soll.

Fehlermeldungen lassen sich selbstverständlich nicht unterdrücken.

ERZEUGE/PRØTØKØLL

formal:

$$\langle \text{Wertzuweisung PRØTØKØLL} \rangle ::= [\text{PRØTØKØLL=}] \left\{ \begin{array}{l} - \\ \langle \text{Angabe} \rangle [' \langle \text{Angabe} \rangle]^\infty \end{array} \right\}$$
$$\langle \text{Angabe} \rangle ::= \&Z \mid A \mid KØ \mid KW \mid Ø$$

Beispiel:

PRØT.=&Z'KW

3.2. Einführung

Die Aufgaben des Kommandos seien zunächst anhand seiner einzelnen Spezifikationen vorgestellt:

1. ZIEL Ein Kommandoaufruf "erzeugt" eine oder mehrere Zeichenketten, die wie unter ZIEL angegeben weiterverarbeitet werden.
- ZIEL=KOMMANDO Eine erzeugte Zeichenkette wird als Kommando-
folge ausgeführt (ähnlich wie beim TU- oder
TUE-Kommando).
- ZIEL=n ($1 \leq n \leq 99999$) Die Zeichenkette wird dem internen Namen *n
zugewiesen.
- ZIEL=TEXT Die Zeichenkette wird ins Ablauf- und Konsol-
protokoll ausgegeben.

Die anderen erlaubten Besetzungen von ZIEL werden unter 4.
behandelt.

2. QUELLE Hier wird als "Ausgangspunkt" der Erzeugung eine Zeichenfolge
angegeben. Im einfachsten Falle (wenn die anderen Spezifikationen
unbesetzt sind) wird sie unmittelbar zum Ergebnis; beispielsweise
führt

```
◇ERZEUGE,KOMMANDO,INFORMIERE
```

zur Ausführung des Kommandos INFORMIERE und

```
◇ERZEUGE, 1, *2
```

weist dem internen Namen *1 den Wert von *2 zu (Achtung:
Dies läßt sich durch *1=*2 nicht erreichen!). Sollen
Zeichen angegeben werden, die in Normalstrings verboten
sind, können ohne weiteres Fremdstrings verwendet werden:

```
◇ERZEUGE, KOMMANDO,/INFORMIERE,ERZEUGE◇/
```

liefert den Aufruf

```
◇INFORMIERE,ERZEUGE .
```

Praktisch ist, daß man beliebig viele Teilwerte (Normal-
und Fremdstrings gemischt) angeben kann, die alle aneinander-
gekettet werden. So läßt sich etwa die folgende Prozedur
schreiben (was mit den Mitteln der Kommandosprache sonst
nicht möglich ist):

```
◇*KOP UND START (DATEI,DMK)
```

```
◇ERZEUGE,KOMMANDO,
```

```
QUELLE=/KOP.,◇/'*DATEI'
```

```
 /W14(EXDKZ(◇/'*DMK'/), -STD-
```

```
◇QUELLE=/STARTE,DATEI=5-◇/'*DATEI'
```

```
◇**
```

Der Aufruf

◇KOP UND START(FRIDA,A)

erzeugt die Kommandos

◇KOP.,FRIDA,W14(EXDKZ(A))

◇STARTE,DATEI=1-FRIDA .

3. TEILWERTE

Oft will man ähnlich wie im letzten Beispiel Teile einer Zeichenkette einzeln angeben, diese aber nicht grundsätzlich immer alle aneinanderketten, sondern nach verschiedenen Gesichtspunkten auswählen lassen. Ein einfaches Beispiel wäre

◇ERZEUGE,1,TEILWERTE=A₁'B₂'C₃'D₄,
KRITERIUM=3-----↑ ,

wodurch ◇*1=C ausgeführt wird.

Welches Element von TEILWERTE ausgewählt wird, hängt außer von KRITERIUM noch von MODUS und anderen Umständen (s.u.) ab.

QUELLE und TEILWERTE lassen sich auch kombinieren:

◇ERZEUGE,KOMMANDO,
QUELLE=/INFORMIERE,◇/
TEILWERTE=UE.'MO.'STARTE,
KRITERIUM=2-----↑

bewirkt

◇INFORMIERE,MO.

4. KRITERIUM

Die Bedeutung der hier gemachten Angaben hängt von MODUS ab.

Im letzten Beispiel war (Voreinstellung!)

MODUS=INDIZIERUNG, deshalb wurde mittels KRITERIUM ein Element von TEILWERTE ausgewählt.

Im MODUS=SCHLEIFE gibt KRITERIUM dagegen an, wie oft etwas geschehen soll.

◇ERZEUGE,KOMMANDO,/DATEI,A◇/
KRITERIUM=3,MODUS=SCHLEIFE

kreiert der Reihe nach die Dateien A(1.0), A(2.0) und A(3.0) (falls noch keine Datei A existierte). Außer einer Anzahl können auch die Abbruchkriterien FEHLER und ERFOLG angegeben werden; was im obigen Beispiel bei KRITERIUM=FEHLER geschehen würde, möge sich der Leser selbst ausmalen.

5. MODUS

Außer den schon erwähnten Modi INDIZIERUNG und SCHLEIFE gibt es noch ANZAHL; hier muß das Ziel ein interner Name sein, dem schlicht die Anzahl der Elemente von TEILWERTE zugewiesen wird:

```
◇ERZEUGE,10,MODUS=ANZAHL,  
TEILWERTE=A'B'15'/EMMA,?'/'◇/'Z3
```

bewirkt die Zuweisung

```
◇*10 = 5 .
```

6. MAL

Hier kann ein beliebiges Zeichen als "Mal" besonders ausgezeichnet werden. Tritt in der QUELLE (oder in den TEILWERTEN) ein Mal auf, so geschieht immer etwas, was von dem unmittelbar darauf folgenden Zeichen abhängt:

Einerseits funktioniert ein Mal wie ein codeabhängiges Fluchtsymbol;

<Mal><Buchstabe>... : Das Mal leitet ein Kommando ein.

<Mal><Ziffer>³... : Es liegt eine Ersatzdarstellung wie z. B. ◇124 vor.

Damit lassen sich also auch ganze Kommandofolgen erzeugen:

```
◇ERZEUGE,KOMMANDO,MAL=§,QUELLE=/  
§EINSCHLEUSE,NAME=GEK§218NSTELT,LFD  
§DRUCKE,,GEK§218NSTELT§ABMELDE◇/
```

Andererseits kann mit Malen erreicht werden, daß QUELLE und TEILWERTE auf kompliziertere Art verarbeitet werden; dies sei zunächst nur angedeutet:

Eine in <Mal>+ ...<Mal>+ eingeklammerte Zeichenfolge wird als arithmetischer Ausdruck aufgefaßt, der berechnet werden soll. Damit sind auf Kommandoebene kleinere Rechnungen möglich:

```
◇*1 = 15
```

```
◇ ERZEUGE,1,MAL=§,QUELLE=§+'*1'+5§+
```

ergibt

```
◇*1 = 20.
```

Ein "Platzhalter" der Form <Mal>!<bel. Zeichen> oder auch <bel. Zeichen><Mal>; zeigt eine Stelle an, an der ein Teilwert von TEILWERTE eingefügt werden soll. (Wie ein Beispiel zeigte, wird, wenn keine Platzhalter auftreten, nur am Ende der QUELLE noch etwas angefügt.

7. PROTOKOLL Diese Spezifikation steuert die Protokollierung;
 ihre Besetzung hat auf den Ablauf der Erzeugung keinen Einfluß.

Es sei noch bemerkt, daß sich außer den Spezifikationsnamen auch die folgenden Spezifikationswerte abkürzen lassen:

KOMMANDO, BEREICH(s.u.), ERFOLG, FEHLER, ANZAHL, INDIZIERUNG, SCHLEIFE

Davon wird im folgenden öfters Gebrauch gemacht werden.

3.3. Wie entsteht ein Ergebnis?

1. Zunächst werden alle unter QUELLE angegebenen Teilwerte aneinandergesetzt, im Falle von Fremdstrings natürlich ohne / und \diamond / .

(Soll das Ergebnis ein Fremdstring sein, muß also beispielsweise geschrieben werden:

\diamond ERZ.,1, //A,B, \diamond / ;

der zweite Schrägstrich leitet beim Ergebnis den Fremdstring ein.)

2. Die so entstandene Zeichenkette wird nun zeichenweise von links nach rechts daraufhin untersucht, ob eine der folgenden Zeichenfolgen vorliegt:

- 2.1. <Mal><Ziffer><Ziffer><Ziffer> (Ersatzdarstellung)
- 2.2.1. <Mal>!<Zeichen \neq Mal> (Platzhalter 1. Art)
- 2.2.2. <Zeichen \neq Mal><Mal>; (Platzhalter 2. Art)
- 2.3. <Mal>+<Zeichenfolge ohne Male><Mal>+ (Formel)
- 2.4. <Mal>?<Zeichenfolge ohne Male><Mal>? (Frage)
- 2.5.1. <Mal>[(NF-Klammer)
- 2.5.2. <Mal>] (F-Klammer)

Liegt eine Ersatzdarstellung (2.1) vor, so werden diese vier Zeichen durch das Zeichen mit dem angegebenen Zentralcodewert ersetzt; <Mal> 124 wird also zum Π .

Hier sei auf die wichtige Tatsache hingewiesen, daß Ersatzdarstellungen, Platzhalter, Formeln, Fragen, NF- und F-Klammern immer erst dann erkannt werden, wenn deren letztes Zeichen untersucht wird. Es wird also beim Auftreten eines Mals nicht "nach rechts" gesehen, ob noch drei Ziffern kommen, vielmehr wird bei Vorfinden einer Ziffer nachgesehen, ob sie die letzte einer Ersatzdarstellung ist. Dadurch ist folgendes möglich:

<Mal><Mal>177<Mal>178<Mal>180

wird zunächst zu

<Mal> 1 2 4

und dann zu

Π .

Gleiches gilt sinngemäß für Platzhalter usw..

Ein Platzhalter (2.2) kann wahlweise die Form

<Mal>!<Zeichen ≠ Mal>

oder

<Zeichen ≠ Mal><Mal>;

haben (dabei soll ; an ein umgedrehtes ! erinnern).

Jeder Platzhalter unterbricht die zeichenweise Interpretation der Kette: Er wird sofort ersetzt durch einen Teilwert von TEILWERTE, und dieser wird zunächst auf weitere Ersatzdarstellungen, Platzhalter u.s.f. untersucht; danach wird dann in der Ausgangskette fortgefahren.

Das<Zeichen ≠ Mal>dient der Benennung der Platzhalter:

Stets werden gleich benannte Platzhalter durch gleiche Teilwerte ersetzt - durch welche, das hängt ab von MODUS und evtl. KRITERIUM. Im MODUS = SCHLEIFE wird der erste gefundene Platzhalter durch den ersten Teilwert ersetzt, jeder weitere mit einer bislang noch nicht aufgetretenen Benennung durch den jeweils nächsten Teilwert:

```
◇ERZ.,KOMM.,MOD.=SCHL.,MAL=$,  
Q.=/$EINSCHL.,$!D.$!N,LFD$DRUCKE,,,$!N◇/  
TEILW.=FRIDA'BERTA
```

erzeugt die Kommandofolge

```
◇EINSCHL.,FRIDA.BERTA,LFD◇DRUCKE,,BERTA
```

Ist MODUS=INDIZIERUNG, wird dem Platzhalter zunächst nicht ein Teilwert von TEILWERTE zugeordnet, sondern einer von KRITERIUM. Dieser muß eine natürliche Zahl sein und gibt die Position des einzusetzenden Teilwertes von TEILWERTE an:

```
◇ERZ.,KOMM.,MOD.=IND.,MAL=$,  
Q.=/$DRUCKE,,,$!N◇/  
T.=A'B'C'D'E.F(3.0)'G-H,  
KRIT.=5
```

ergibt

```
DRUCKE,,E.F(3.0) ,
```

weil E.F(3.0) der 5. Teilwert von TEILWERTE ist.

Es ist erlaubt, daß die TEILWERTE selbst wieder Platzhalter enthalten; dies führt allerdings leicht zu Unübersichtlichkeiten:

```
◇ERZ.,KOMM.,MOD.=SCHL.,MAL=$,  
Q.=/$DRUCKE,,,$!A,1KOP◇/  
T.=,$!D.$!N'DB'DATEI-$!P'PW
```

ergibt (man versuche das nachzuvollziehen) tatsächlich

```
◇DRUCKE,,DB.DATEI-PW,1KOP ;
```

so etwas wie

```
◇ERZ.,1,MOD.=SCHL.,MAL=$
Q.=$!1$!1,T.=A$!1A
```

sollte man jedoch lieber nicht probieren

Ist das Ende der Zeichenfolge erreicht, ohne daß überhaupt Platzhalter gefunden wurden, wird ein impliziter Platzhalter ganz am Ende angefügt:

```
◇ERZ.,KOMM.,/UEB.,◇/,A
```

wirkt wie

```
◇UEB.,A .
```

Dabei darf QUELLE auch gänzlich unbesetzt sein:

```
◇ERZ.,1,MOD.=IND.,KRIT. =*2,
TEILW.=A'B'C'D'
```

weist dem internen Namen beispielsweise ein B zu, falls *2 mit 2 besetzt war.

Auch einzelne Teilwerte von TEILWERTE dürfen leer sein. Dies erreicht man - da "undefiniert" als Teilwert verboten ist - durch die Angabe leerer Fremdstrings (etwa A'◇/'B).

Formeln (2.3) sind in <Mal> + ...<Mal> + eingeschlossene Zeichenfolgen.

Sie werden als arithmetische Ausdrücke aufgefaßt und nach Berechnung durch das Ergebnis ersetzt:

```
◇ERZ.,15,/$1$+2*12$+◇/,MAL=$
```

führt über das Zwischenergebnis \$124 zu der Zuweisung

```
◇#15 = 124 .
```

Ähnlich verhält es sich mit in <Mal> ? ... <Mal> ? eingeschlossene Fragen (2.4); diese werden als Operateurfrage auf der Kontrollschreibmaschine ausgegeben und durch die vom Operateur gegebene Antwort ersetzt. Dies ist für RZ-Dienstprogramme nützlich, wie ein Beispiel andeuten möge:

```
◇ERZ.,KOMM.,MAL=$,QUELLE=/
$LFVERWALTE,KONSERVIERE,TRAEGER=MB($?
Gib EXDKZ für LFD-Konserve:$?)
```

Gibt der Operateur auf die Anfrage etwa LFDO29 an, wird das Kommando

```
◇LFVERWALTE,KONSERVIERE,TRAEGER=MB(LFDO29)
```

ausgeführt.

Eine ganz andere Funktion haben die NF- und F-Klammern (2.5). Sie müssen stets ein vollständiges "Klammergebirge" bilden.

Innerhalb von <Mal>[und <Mal>] werden Male nur als solche erkannt, wenn sie weitere NF- und F-Klammern einleiten; in allen anderen Fällen werden Male innerhalb von <Mal>[und <Mal>] wie gewöhnliche Zeichen behandelt.

Die jeweils äußerste Schicht von NF- und F-Klammern wird entfernt.

◇ERZ.,1,\$(\$ \$),MAL=\$

liefert also beispielsweise

◇*1 = \$.


Diese Klammern sind insbesondere dann von Bedeutung, wenn eine Zeichenkette für einen späteren, erneuten ERZUGE-Aufruf vorbereitet werden soll (Beispiele hierzu folgen weiter unten). Außerdem erlauben sie die Verwendung des vollen Zeichensatzes ohne Nebenwirkungen: Kennt man etwa die Besetzung von *100 nicht, möchte aber das Wirksamwerden von Malen ausschließen, kann man schreiben

...,QUELLE=\$['*100'\$],MAL=\$,...

(was nur dann schiefgeht, wenn *100 die Zeichenfolge \$] enthält.)

3.4. Wie werden die TEILWERTE ausgewählt?

1. Im MODUS=SCHLEIFE werden die Teilwerte von TEILWERTE von links nach rechts den verschiedenen Platzhaltern zugeordnet. Hat ein Platzhalter allerdings die gleiche Benennung wie ein anderer, früher gefundener, so wird ihm auch der gleiche Teilwert wie jenem zugeordnet:

◇ERZ.,1,\$!A \$!B \$!A \$!C \$!A,TEILWERTE=0'1'2

 MOD.=SCHL.,MAL=\$

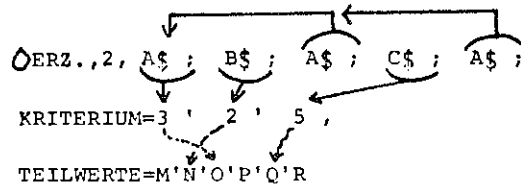
liefert

◇*1=01020 .

Es werden also immer genau so viele Teilwerte benötigt, wie verschieden benannte Platzhalter vorkommen. Ausnahme: Kommt überhaupt kein Platzhalter vor, so wird ein Teilwert benötigt; er wird ans Ende der QUELLE angehängt.

2. Im MODUS=INDIZIERUNG werden den Platzhaltern zunächst nicht Teilwerte von TEILWERTE, sondern von KRITERIUM zugeordnet. Auch hier gehören zu gleich benannten Platzhaltern gleiche Teilwerte.

Die Teilwerte von KRITERIUM wirken als Indizes, die auf einzelne Teilwerte von TEILWERTE zeigen. (Im folgenden Beispiel werden zur Abwechslung die Platzhalter 2. Art verwendet.)



ergibt

◇*2=ONOOO .

In diesem Modus werden genau so viele Kriterien benötigt, wie verschieden benannte Platzhalter auftreten (bzw. genau ein Kriterium, falls überhaupt keiner gefunden wird); dagegen ist die Anzahl der Teilwerte von TEILWERTE beliebig.

Im MDOUS=INDIZIERUNG kann man sich die Liste der TEILWERTE als mit dem letzten Element beliebig verlängert vorstellen: KRITERIUM=n+k ($k \geq 0$) wie KRITERIUM=n, wenn nur n TEILWERTE angegeben werden.

3.5. Was geschieht mit dem Ergebnis?

1.1. Im Falle ZIEL=KOMMANDO wird der Entschlüsselung in der Vorrangstufe gestartet und das Ergebnis als Kommandofolge ausgeführt. Vor das Ergebnis wird dabei stets ein Fluchtsymbol gesetzt, weitere Fluchtsymbole können aus Malen hervorgehen.

Ist das Ergebnis die leere Zeichenfolge, so unterbleibt der Entschlüsselungstart; auch dies gilt als fehlerfreie Ausführung.

1.2. Im Falle ZIEL=n ($1 \leq n \leq 99999$) wird vor dem Ergebnis ein

◇*n=

angefügt; dann wird ebenfalls der Entschlüsselung gestartet.

Ist das Ergebnis die leere Zeichenfolge, so wird

◇*n=<"undefiniert">

zugewiesen; auch dies gilt als fehlerfreie Ausführung.

1.3. Im Falle ZIEL=BEREICH muß das Ergebnis der Name eines (rechenzentrumsspezifischen) Problemkreises sein; es werden dann die zu diesem Problemkreis gehörigen Kommandos ausgeführt. (Näheres unter 4.)

1.4. Ist ZIEL=TEXT, so wird das Ergebnis ins Ablauf- und Konsolprotokoll gedruckt.

2. Ist MODUS=INDIZIERUNG, so geschieht das unter 1.1. ... 1.3. geschilderte genau einmal. Im MODUS=SCHLEIFE ist dies jedoch anders. Hier werden die Angaben unter KRITERIUM nicht als Indizes für die TEILWERTE, sondern als Abbruchkriterium für wiederholte Ausführung interpretiert. Besonders sinnvoll ist dies bei ZIEL=KOMMANDO: Ist KRITERIUM=n, so erfolgt n-malige Ausführung; neben natürlichen Zahlen n sind die Abbruchkriterien FEHLER und ERFOLG erlaubt:


```

◇ERZ.,KOMM.,MOD.=SCHL.,KRIT.=FEHLER,MAL=§
  QUELLE=/$UMBENENNE,A'(9999.99)
    $LOESCHE,DATEI=A(9999.99)

```

löscht alle (!) Dateien mit Namen A und beliebiger G-V-Nummer - die mit der höchsten zuerst.

```

◇ERZ.,KOMM.,MOD.=SCHL.,KRIT.=ERF.,MAL=§,
  QUELLE=/EINSCHLEUSE,A,,LFD

```

wird eine Datei einschleusen, sobald sie vorhanden und nicht von jemand anders angemeldet ist. (Da die Versuche unmittelbar aufeinander folgen, wird so allerdings u.U. viel Rechenzeit verbraucht - besonders, wenn man sich beim Dateinamen verschreibt!)

Im MODUS=ANZAHL werden dagegen QUELLE und KRITERIUM überhaupt nicht angesehen. ZIEL muß ein interner Name sein, dem lediglich die Anzahl der Teilwerte von TEILWERTE zugewiesen wird.

3.6. Wann entstehen mehrere Ergebnisse?

Für ein Ergebnis werden ein Teilwert von ZIEL und so viele Teilwerte von TEILWERTE (im MODUS=SCHLEIFE) bzw. von KRITERIUM (im MODUS=INDIZIERUNG) benötigt, wie verschieden benannte Platzhalter vorkommen (bzw. genau einer, wenn gar kein Platzhalter vorkommt). Im MODUS=SCHLEIFE wird ferner für ein Ergebnis noch ein Teilwert von KRITERIUM benötigt. Wurden nun irgendwo mehr Werte als nötig angegeben, so entsteht ein zweites und evtl. noch weitere Ergebnisse aus den bisher unbenötigten Werten. Sind nicht alle Listen zugleich abgearbeitet, so kann man sich die kürzeren als mit dem letzten Element so weit wie nötig aufgefüllt denken.

Ein "abstraktes" Beispiel möge zunächst den allgemeinen Fall erläutern; die folgenden Beispiele zeigen dann, wie man - ohne daß die Sachlage unübersichtlich wird - diese Tatsache für viele praktische Anwendungen ausnutzen kann.

```

◇ERZ.,1'2'KOMM.($!ARB$),1'2'3'4'D'PR.'/◇/,
  MOD.=SCHL.,MAL=§,KRIT.=1'1'3'ERF.

```

Das erste Ergebnis ist

```
1 R 2 ;
```

es wird dem internen Namen *1 (1.ZIEL=1) genau einmal (1.KRITERIUM=1) zugewiesen.

Da noch nicht alle Listen abgearbeitet sind, entsteht ein zweites Ergebnis unter Hinzuziehung der nächsten TEILWERTE:

```
3 R 4 .
```

Es wird (wie das 2. ZIEL und das 2. KRITERIUM angeben) dem internen Namen *2 einmal zugewiesen. Danach entsteht ein drittes Ergebnis, nämlich

DRPR. ,

welches (vgl. 3. ZIEL und 3. KRITERIUM) dreimal als Kommando ausgeführt wird. Die ZIEL-Liste ist jetzt abgearbeitet, die anderen jedoch noch nicht; deshalb beginnt eine vierte Erzeugung. Die Quelle enthält zwar zwei verschieden benannte Platzhalter, es ist aber nur noch ein Teilwert von TEILWERTE da (nämlich die leere Zeichenkette); dieser muß also für beide Platzhalter genommen werden, womit das vierte Ergebnis

R

lautet.

Das letzte ZIEL war KOMMANDO, das vierte KRITERIUM ist ERFOLG; also wird R so lange ausgeführt, bis dies ohne Fehler gelingt.

Danach ist kein Teilwert mehr unbenutzt und die Erzeugung ist beendet.

Nun einige praxisnähere Beispiele:

1. Der Inhalt mehrerer Dateien soll ausgedruckt werden.

```
◇ERZ.,KOMM.,/DRUCKE,,◇//,DAT1'DAT2'DAT3,MOD.=SCHL.
```

(Ist kein KRITERIUM angegeben, so wird stets 1 eingesetzt.)

2. Verschiedene Dateien sollen in verschiedenen Codes auf Karten gestanzt werden.

```
◇ERZ.,KOMM.,/$STANZE,KS-!C,D$;◇/,  
TEILW.=KC2'DAT1'KC1'!AT2'BIN'BINDAT,  
MOD.=SCHL.,MAL=$,KRIT.=2'3
```

Die hierdurch ausgeführten Kommandos lauten

```
◇STANZE,KS-KC2,DAT1  
◇STANZE,KS-KC2,DAT1  
◇STANZE,KS-KC1,DAT2  
◇STANZE,KS-KC1,DAT2  
◇STANZE,KS-KC1,DAT2  
◇STANZE,KS-BIN,BINDAT  
◇STANZE,KS-BIN,BINDAT  
◇STANZE,KS-BIN,BINDAT
```

3. Verschiedene interne Namen sollen umbesetzt werden.

```
◇ERZ.,1'2'5'18'100,TEILW.=*100'*18'*5'*2'*1,  
MOD.=SCHL.
```

Dies bewirkt die Zuweisungen

```

◇*1= <alter Inhalt von *100>
◇*2= <alter Inhalt von 18>
:
◇*100= <alter Inhalt von *1> ;

```

dies läßt sich bekanntlich durch die Kommandos

```

◇*1= *100
:

```

nicht erreichen, da hierbei nicht Inhalte übertragen, sondern Verweise umgesetzt werden.

4. Der RZ-spezifische Bereich ZEIGE möge die Definition eines Kommandos ZEIGE enthalten. Dieses Kommando soll zuerst aktiviert und dann auch ausgeführt werden.

```

◇ERZ.,BEREICH'KOMMANDO,ZEIGE

```

5. Mehrere Bereiche sollen gleichzeitig aktiviert werden.

```

◇ERZ.,BEREICH,TEILW.=BER1'BER2'BER3

```

- 6.1. Ein interner Name soll eine Liste von Dateinamen enthalten, die sich bei Bedarf verlängern läßt.

Da diese Liste Apostrophe enthalten muß, wird sie als Fremdstring verwaltet.

Unter der Voraussetzung

```
*100 = /DATEI1
```

ist es möglich, mit der Prozedur

```
◇*ANFUEGE (DATEI)
```

```
◇ERZ.,100, //◇/'*100'/'◇/'*DATEI
```

die Liste beliebig oft zu verlängern.

(*100 muß ein Fremdstring sein, da sonst (Wirkung von QUELLE!) alle Teilwerte beim nächsten ERZUEGE-Aufruf zu einer Kette verknüpft würden. Damit bei jeder Erzeugung wieder ein Fremdstring entsteht, muß vorn ein Schrägstrich angefügt werden.)

- 6.2. Alle in der Liste aufgeführten Dateien sollen gesichert werden.

Zunächst wird aus dem Fremdstring *100 eine Liste von Dateibezeichnungen erzeugt, die auf *200 abgelegt werden soll:

```
◇ERZ.,200,*100
```

Das Sichern geschieht dann mittels

```
◇ERZ.,KOMM.,MOD.=SCHL.,MAL=,$,TEILW.=*200,
```

```
QUELLE=/SICHERE,D$;,MB(OOOO01)1.9999
```

(Ist man ganz vorsichtig, kann man noch KRITERIUM=ERFOLG angeben.)

3.7. Protokollierung

Im Normalfalle (PROTOKOLL="undefiniert") erfolgt abgesehen von Fehlermeldungen und Warnungen lediglich der Druck der üblichen Endmeldung mit Angabe der Verbrauchten Rechenzeit ins Ablauf- und Konsolprotokoll.

Ist diese Meldung unerwünscht, kann PROTOKOLL=&Z ("unterdrückte Zeitmeldung") angegeben werden. Dies ist z. B. dann nützlich, wenn ein ERZEUGE-Kommando Teil einer Prozedur ist, deren Bestandteile den Benutzer nicht interessieren sollen.

Fehlermeldungen sind selbsterklärende Texte, die vor einem Abbruch der Kommandoausführung ausgegeben werden, wenn eine Fortsetzung nicht mehr möglich oder sinnvoll ist.

In einigen Fällen werden Warnungen ausgegeben; diese führen nicht zum Abbruch.

Im Gegensatz zu Fehlermeldungen lassen sich alle Warnungen durch PROTOKOLL=KW ("keine Warnungen") unterdrücken.

Eine ausführliche Protokollierung aller entstehenden Ergebnisse mit den zugehörigen Zielen und Wiederholungskriterien läßt sich durch PROTOKOLL=A ("ausführlich") erreichen. Die Angabe PROTOKOLL=O dient zur Einkreisung von Fehlern; sie bewirkt Ausgabe der Versorgung des Kommandos, einiger interner Versorgungsblöcke und Startsätze sowie aller Zwischenergebnisse.

Die Angaben &Z, KW, A und O können als Teilwerte beliebig kombiniert werden.

Soll das A- bzw. O-Protokoll auch auf Konsole ausgegeben werden, ist zusätzlich KO anzugeben. (Fehlermeldungen und Warnungen werden natürlich im Gespräch immer ins Konsolprotokoll geschrieben.)

4. SCHNITTSTELLEN ZUM PROGRAMMIERSYSTEM

Dieses Kapitel behandelt einige mit der Implementierung zusammenhängende Fragen und wendet sich hauptsächlich an das Rechenzentrum.

4.1. BEREICH

Soll die Aktivierung von Bereichen möglich sein (ZIEL=BEREICH), muß eine wie folgt deklarierte Datei in der LFD vorhanden sein:

```
DATEI,KOMMANDOS(9999.99),RAM-G,U...,U800,LFD(RZ)
```

Diese Datei muß nach RB&QUELLHALT-Konventionen aufgebaut sein (vgl. Arbeitsbericht Nr. 7305); das heißt:

Die Sätze 2 - 4999 enthalten ein Inhaltsverzeichnis. Jeder Bereich wird darin durch einen Satz vertreten, der in Spalten 7 - 18 den Namen des Bereichs (linksbündig) enthält und in Spalte 38 - 50 eine sechsstellige Satznummer, ein Minuszeichen und eine weitere sechsstellige Satznummer.

Im Falle ZIEL=BEREICH wird diese Datei zum Lesen eingeschleust, dann werden alle Bereichsnamen im Inhaltsverzeichnis mit dem Ergebnis der Erzeugung verglichen. (Dieses Ergebnis darf auch mit Punkt abgekürzt sein.) Bei der ersten Übereinstimmung werden die im angegebenen Dateibereich stehenden Kommandofolgen mit TU (BO&TUE) ausgeführt. Zusätzlich wird im Anschluß daran immer die in Satz 5010 stehende Kommandofolge ausgeführt. Diese sollte lauten:

```
WAHLSCHALTER,WS1,- SPRINGE,ENDE,(FE2) WAHLSCHALTER,-,WS1
```

(BO&ERZEUGE fragt anschließend WS1 ab um festzustellen, ob ein Fehler aufgetreten war.)

Das erste Zeichen in Satz 1 wird in der gesamten Datei als Fluchtsymbol aufgefaßt.

Die Datei KOMMANDOS könnte also etwa wie folgt aussehen:

```
000001 #=      I N H A L T S V E R Z E I C H N I S      :
000002 #=      ,LEERBEREICH ,#TUE,KOMMANDOS      , 5000- 5010
000003 #=      ,ABKUERZUNGEN,#TUE,KOMMANDOS      , 5800- 6100
000004 #=      ,DIATRACE      ,#TUE,KOMMANDOS      , 6200- 7500

004999 # ENDE
005000 #=      ,LEERBEREICH,      VOM 09.01.75
005010 #WAHLSCHALTER,WS1,-#SPRINGE,ENDE,(FE2)#WAHLSCHALTER,-,WS1
005800 #=      ,ABKUERZUNGEN,      VOM 09.01.74
005810 #*AB (G,M,P,F,D)
005820 #BINAERAUS,GERAET=*G,MQ=*M,PROGRAMM=*P, FREMOSYS=*F, DATEI
=*D ***
005830 #*EB(I) #BINAEREIN,INF.*I ***
005840 #*BE(PIB,T) #PIBVERL.,*PIB,*T,-STO,-,STO,-,STO,-,MODE'PRO'KI
***
005850 #*T(BE)=LFD
005860 #*BA(PIB,T) #PIBV.,*PIB,*T,*T #PIBV.,*PIB,-STO,-,*T,-STO,-,
STO-,
005870 KI'PRO'MDE***
005880 #*T(BA)=LFD
005900 #*SW (W) #WAHLSCHALTER, SETZE=*W ***
006000 #*LW (W) #WAHLSCHALTER, LOESCHE=*W ***
006100 #*DPL #ORPROTO KILL,ZUSTAND=LOESCHE ***
006200 #=      ,DIATRACE,      VOM 09.01.74
006300 #BIBANM.,HELPER'RZ,TR.=LFD,PROT.=-
006400 #STARTE,F0&DIATRACE,DEFINIERE
```

4.2. Formeln

Alle in \$+... \$+ eingeschlossenen Formeln werden dem Operator AIDA (vergleiche Arbeitsbericht 7705) zur Interpretation übergeben; zulässig sind somit beliebige AIDA-Programme.

Die Informationsübergabe erfolgt in einer sequentiellen Kernspeicher-Datei.

Sollen Formeln interpretiert werden, muß also der Operator AIDA zur Verfügung stehen.

Bisher erschienene Arbeitsberichte des Rechenzentrums
der Ruhr-Universität Bochum

- Nr. 7101: K.-H. Mohn, M. Rosendahl, H. Zoller
AIDA; eine Dialogsprache für den TR 440 (vergriffen)
- Nr. 7102: K.-H. Mohn, M. Rosendahl, H. Zoller
AIDA, ein Dialogsystem und seine Implementierung in ALGOL (vergriffen)
- Nr. 7103: K.-H. Mohn, M. Rosendahl, H. Zoller
AIDA, Manual für den Benutzer (vergriffen)
- Nr. 7104: 4. Jahresbericht des Rechenzentrums (Juni 1970 bis Juni 1971)
- Nr. 7105: H. Wupper
WR 1802 - Ein einfaches Band-Betriebssystem für einen mittleren Rechner
- Nr. 7201: H. Windauer
Existenzsätze zur $(0, 1, \dots, R-2, R)$ - Interpolation
- Nr. 7202: W. Schelongowski
DIATRACE - Ein System zur interaktiven Assemblerprogrammierung
- Nr. 7203: M. Jäger, M. Rosendahl, R. Staake
Einführung in die Listenverarbeitung anhand der Dialogsprache AIDA
- Nr. 7204: R. Mannshardt, P. Pottinger
Einführung in die Benutzung des Teilnehmer-Rechensystems TR 440 in der RUB (vergriffen)
- Nr. 7205: 5. Jahresbericht des Rechenzentrums (1.7.1971 bis 30.6.1972)
- Nr. 7206: M. Rosendahl
BOGOL-TAS, ein Weg zur systemnahen Programmierung in ALGOL am TR 440
- Nr. 7207: W. Stark
ILW, Programmsystem zur Berechnung des Instationären Ladungswechsels von
Verbrennungskraftmaschinen (Modulbeschreibung und Eingabekonventionen)
- Nr. 7208: W. Stark
ILW, Programmsystem zur Berechnung des Instationären Ladungswechsels von
Verbrennungskraftmaschinen (Regelmechanismus und Berechnung der Rohrströmung)
- Nr. 7209: H. Ehlich
Anregung und Kritik zum Betriebs- und Programmiersystem der TR 440
- Nr. 7210: M. Rosendahl
BOGOL-STRING, eine flexible Zeichenkettenverarbeitung in ALGOL 60
- Nr. 7211: H. Camici, H. Claus, H. Ehlich, D. Kipp
Arbeitsbericht über ein Programm zur Haushaltsführung
- Nr. 7301: R. Mannshardt, K.-H. Mohn, H. Münch, P. Pottinger
Einführung in die Benutzung des Teilnehmer Rechensystems TR 440
2. geänderte Auflage (vergriffen)

- Nr. 7302: K.-H. Mohn
Über einige Anwendungen des Computers in der Medizin
- Nr. 7303: R. Buchmann
BODAT, ein schnelles und platzsparendes System zur Datenmanipulation und
-speicherung in ALGOL 60 und FORTRAN
- Nr. 7304: M. Hauenschild
Ansätze zur komplexen Kreisarithmetik
- Nr. 7305: R. Buchmann
RB&QUELLHALT, ein TR440-Datenbanksystem zur platzsparenden Quellhaltung auf
Datenträgern mit direktem Zugriff (LFD, WSP)
- Nr. 7306: 6. Jahresbericht des Rechenzentrums (1.7.1972 bis 31.12.1973)
- Nr. 7401: R. Buchmann
Der Systemoperator BO&BS30P
Messungen und Steuerungen des Betriebssystems auf Operatorebene
- Nr. 7402: R. Mannhardt
Herleitung und Prüfung spezieller Runge-Kutta-Verfahren mit einem impliziten Rechenschritt
- Nr. 7403: R. Buchmann, H. Wupper
Unzulänglichkeiten des TR 440 Programmiersystems und ihre Umgehung
- Nr. 7404: R. Green, K.-H. Mohn
Quellbezogene FORTRAN Optimierungen für den Compiler des TR 440
- Nr. 7405: R. Buchmann
BODAT, ein schnelles und platzsparendes System zur Datenmanipulation und
-speicherung in ALGOL 60 und FORTRAN (2., ergänzte Auflage)
- Nr. 7501: R. Buchmann
Zur Theorie der Montage von Programmmoduln
- Nr. 7502: 7. Jahresbericht des Rechenzentrums (1.1. bis 31.12.1974)
- Nr. 7503: H.-D. Sander
BOTRAN, eine Fortran Spracherweiterung durch Code-Prozeduren
- Nr. 7504: W. Schelongowski
DIATRACE - Ein System zur interaktiven Assemblerprogrammierung
- Nr. 7505: Camici, Prof. Dr. Ehlich, Schürmann
Über ein Programm zur Material- und Vervielfältigungs-Abrechnung
- Nr. 7506: Camici, Prof. Dr. Ehlich, Herrmannies
Über ein Programm für die Telefonabrechnung einer Nebenstellenanlage
- Nr. 7507: Camici, Prof. Dr. Ehlich, Kipp
Über ein Programm zur Haushaltsführung
- Nr. 7508: Camici, Cipa, Prof. Dr. Ehlich
Bericht über ein Programm zu Verwaltung der Studentendaten
- Nr. 7509: J. Riege
Zur mehrdimensionalen Spline-Interpolation bezüglich beliebiger linearer Funktionale

- Nr. 7601: H. Ehlich, J. Riege u. K.-H. Schloßer
Ein Programmsystem zur Ausleihverbuchung und interaktiven Rechnerunterstützung in
der allgemeinen Buchbestandsverwaltung
Teil 1: Offline-System
- Nr. 7602: M. Rosendahl
BOGOL-STRING, eine flexible Zeichenkettenverarbeitung in ALGOL60
2. erweiterte und geänderte Version
- Nr. 7603: R. Buchmann, M. Rosendahl
BOGOL-TAS, eine Spracherweiterung von ALGOL60 durch Codeprozeduren
zur Systemprogrammierung
- Nr. 7604: R. Buchmann
AUFBEREITE, ein universell einsetzbarer Dateiänderungsoperator für verschiedene
Datei- und Dialoggerätypen und/oder Betriebsarten
- Nr. 7605: 8. Jahresbericht des Rechenzentrums (1. Januar bis 31. Dezember 1975)
- Nr. 7606: R. Buchmann, H. Wupper
BO&ZEICHNE, Vorschlag zur geräteneutralen Graphik
- Nr. 7701: Camici, Ehlich, Kipp, Wiedemann
Inventarisierungsprogramm
- Nr. 7702: K.A. Görg, W. Stark, R. Wojcieszynski
Verfahren zur Berechnung der Strömung durch eine Drosselstelle ohne Speicherwirkung
- Nr. 7703: H. Wupper
ALGOL68 Eine Einführung in die Programmierung für Anfänger
- Nr. 7704: Camici, Prof. Dr. Ehlich, Volmer, Wiedemann
Aufbau und Verwendung der Personaldatei der Ruhr-Universität Bochum
- Nr. 7705: M. Rosendahl
AIDA, Handbuch für den Benutzer
- Nr. 7706: V. Riedel, K.-H. Schloßer
Ein Programmsystem zur Ausleihverbuchung und interaktiven Rechnerunterstützung
in der allgemeinen Buchbestandsverwaltung
Teil 2: On-line-System
- Nr. 7707: M. Peuser, H. Wupper
Ein geräteneutrales und rechnerneutrales System zur graphischen Ausgabe
- Nr. 7708: H. Wupper
ERZEUGE, eine Erweiterung der Kommandosprache