

RECHENINSTITUT
DER
TECHNISCHEN HOCHSCHULE STUTTGART

FORMATPROGRAMM

fuer die
elektronische Rechenanlage ER 56

Juni 1966

1. Zweck des Programms .

Das Formatprogramm erlaubt die formatgebundene Ausgabe von im Kernspeicher stehenden Daten, Befehlen und Texten direkt ueber den Fernschreiber oder im Fernschreibcode ueber den Stanzer. Das Format wird dabei vom Benutzer ueber einen Formatstring bestimmt, dessen Aufbau unter 4. beschrieben wird. Mit Hilfe des Strings koennen ebenfalls Texte in die Ausgabe eingeblendet werden. Das Formatprogramm ersetzt damit u.a. einen bisher notwendigen Formatstreifen.

2. Aufbau und Organisation des Programms .

Das Programm liegt als Unterprogramm in der Programm-Bibliothek fuer die ER 56 vor. Es benoetigt die KS-Zellen 5980 - 5999 , kann aber ansonsten an eine beliebige Stelle des KS gebracht werden .

Laenge des Programms : ~~630~~ Befehle .

Die Inhalte saemtlicher Indexregister und PM bleiben erhalten. (Ausnahme: Siehe 5. Fehlermeldungen).

Unmittelbar auf den Absprungbefehl ins Formatprogramm folgen im Hauptprogramm 3 Versorgungszellen zur Parameteruebergabe. Der Ruecksprung ins Hauptprogramm erfolgt in die darauffolgende Zelle.

Der Formatstring ist ebenfalls ausser in die Zellen 5980 - 5999 an einen beliebigen Ort des KS zu bringen .

3. Bedeutung der zu uebergebenden Parameter .

Inhalt der 1. Versorgungszelle : AA i b

Inhalt der 2. Versorgungszelle : EA i d

Inhalt der 3. Versorgungszelle : FORM i e

Dabei bedeuten :

AA + (J1) : Anfangsadresse der auszugebenden Daten .

EA + (J1) : Endadresse " " " ,
falls b = 03 , sonst ohne Bedeutung .

FORM + (J1) : Anfangsadresse des Formatstrings; ohne Bedeutung in den
Faellen b = 05 und b = 06 .

Wird i = 9 gewaehlt, so wird zu AA, EA bzw. FORM die um 1 erhoechte Adresse der entsprechenden Versorgungszelle addiert.

e , d : Die zweiziffrige Zahl d wird als α -Zeichen interpretiert und vor der Ausgabe der eigentlichen Information e-mal ausgegeben .

Die Zahl b gibt eine Betriebsart an :

b = 00 : Ausgabe der Daten ab AA + (J1) bis $\omega\omega$ -Wort oder Ende des KS , in dem AA + (J1) liegt .

b = 01 : Ausgabe bis $\omega\omega$ -Wort oder bis Zelle 5979 einschl.

b = 02 : Ausgabe bis KS - Ende .

b = 03 : Ausgabe bis EA + (J1) einschl.

b = 04 : Ausgabe eines einzelnen ab AA + (J1) stehenden Dats .

- b - 05 : Ausgabe eines ab AA + (J1) stehenden α -verschlüsselten Textes bis zu einem Wort mit Vorzeichenstelle $\neq 3$ oder $\neq 6$ oder bis KS- Zelle 5979 einschl.
- b - 06 : Wie bei b - 05 , aber Ausgabe bis zu einem Wort mit Vorzeichenstelle $\neq 3$ oder bis KS-Zelle 5979 einschl.
- b - 07 : Abbruch des Formatprogramms nach c-maliger Ausgabe des Zeichens d .

4. Der Formatstring.

4.1 Aufbau des Formatstrings .

Der Formatstring entspricht dem bisher am Druckerpult benutzten Formatstreifen; er besteht aus einem α -Text und hat die Form

(Spezifikation, Spezifikation,)

Innerhalb des Formatstrings sind Zwischenraeume, Wagenruecklaeufe und Zeilentransporte ausser bei Einfuegungen (s.u.4.4.4.2) bedeutungslos und koennen zur uebersichtlicheren Gestaltung des Strings verwendet werden. Eine Spezifikation dient zur Ausgabe genau eines Dats, sofern sie nicht wiederholt wird (s.u.). Die Abtrennung einzelner Spezifikationen durch Kommata ist nur notwendig, wenn eine Spezifikation mit einem Zahlenwert endet und die naechste mit einem Zahlenwert beginnt.

Als Spezifikation gilt ferner der Ausdruck

m(Spezifikation, Spezifikation,)

Ist $m \neq 1$, so wird der Inhalt der Klammer m-mal wiederholt; fehlt m, so wird die Klammer solange wiederholt, bis der Datenvorrat erschoept ist. Die Spezifikation 2(Spez.1, Spez.2) hat also dieselbe Wirkung wie die Folge von Spezifikationen

Spez.1 , Spez.2 , Spez.1 , Spez.2 .

4.2 Aufbau der Spezifikationen

In Folgenden stehen in den einzelnen Spezifikationen kleine Buchstaben an Stelle von Zahlen. Alle anderen Buchstaben und Zeichen treten als solche in den Spezifikationen auf. Ein Zwischenraum werde durch $_$ verdeutlicht.

Je nach ihrer Anwendung werden 4 Arten von Spezifikationen unterschieden. Dabei bedeuten einheitlich :

- $i \geq 0$: Zahl der Dezimalstellen vor dem Dezimalpunkt.
- $0 \leq d < 50$: Zahl der Dezimalstellen nach dem Dezimalpunkt.
- $n > 0$: Zahl der Wiederholungen fuer diese eine Spezifikation.

4.2.1 : Ausgabe von Daten :

Jedes Dat wird rechtsbuenig in einem Feld der Feldweite $w \geq 0$ ausgegeben. Die Daten werden entsprechend der gewünschten Stellenzahl gerundet. Positive Vorzeichen sowie fuhrrende Nullen vor einem Dezimalpunkt werden unterdrueckt. Der Dezimalpunkt selbst wird im Falle $d=0$ unterdrueckt.

4.2.1.1 : Gleitkomma - Ausgabe :

Bed. : Ein in der Betriebsart GK mit dem Befehl 32 abgespeichertes Dat wird als Gleitkomma - Zahl ausgegeben.

Spez.: $iFnGw.d$

4.2.1.2 : Festkomma - Ausgabe 1 :

Bed. : Ein in der Betriebsart GK mit dem Befehl 32 abgespeichertes Dat wird als Festkomma - Zahl ausgegeben.

Spez.: $nFw.d$

4.2.1.3 : Festkomma - Ausgabe 2 :

Bed. : Ein in der Betriebsart FKN mit Befehl 42 abgespeichertes Dat wird als Festkomma - Zahl ausgegeben.

Spez.: 1PnW.d

4.2.1.4 : Festkomma - Ausgabe 3 :

Bed. : Ein in der Betriebsart FKL mit Befehl 32 abgespeichertes Dat wird als Festkomma - Zahl ausgegeben.

Spez.: 1PnW.d

4.2.2 Ausgabe von a - Text :

Zur Ausgabe von a - Text bestehen 2 Moeglichkeiten:

4.2.2.1 : Soll ein Dat in einem der unter 4.2.1 angegebenen Formate ausgegeben werden und es steht innerhalb der Ausgabedaten ein a - Text an, kenntlich an einer 3 oder 6 in der Vorzeichenstelle, so wird zunaechst dieser Text ausgegeben.

4.2.2.2 : Die Spezifikation nH bewirkt, dass die im String auf H folgende Folge von n Zeichen unveraendert ausgegeben wird ; dabei werden Wagenruecklauf, Zeilentransport und Zwischenraeume beruecksichtigt .

4.2.3 Ausgabe von Befehlen :

Mit der Spezifikation nBw koennen beliebige Zelleninhalte als siebenstellige ganze Zahlen ausgegeben werden.

4.2.4 Ausgabe von Ausrichtzeichen :

4.2.4.1 : Durch die Spezifikation n/ wird ein Wagenruecklauf mit anschliessenden n Zeilentransporten ausgegeben.

4.2.4.2 : Durch die Spezifikation nX werden n Zwischenraeume ausgegeben.

Bei saemtlichen Spezifikationen koennen fuer 1-0 die Bestandteile 1P und es kann fuer n=1 die Zahl n entfallen.

5. Fehlerausdrucke :

- 5.1 FMT - ENDE SYNT : Im Formatstring tritt ein Nicht-a-Wort auf, bevor dieser beendet ist; in den meisten Faellen wird die Klammerstruktur fehlerhaft sein.
- 5.2 SYNT : Syntaxfehler im Formatstring.
- 5.3 SCHACHT ZU GROSS: Der Schachtelungsgrad ist zu gross. Es ist nur ein Schachtelungsgrad bis maximal 10 zulaessig. Aenderung des Formatprogramms fuer hoehere Schachtelungsgrade : Siehe interne Programmbeschreibung.
- 5.5 D ZU GROSS : $d \geq 50$ ist nicht zulaessig.
- 5.6 W ZU KLEIN : Die Feldweite wurde zu klein gewaehlt.

Nach jedem Fehlerausdruck laeuft das Programm auf den Stop 9000019. Nach Druecken der Taste START werden die noch nicht ausgegebenen Daten mit Hilfe des Strings ((4G:6.8/)) ausgegeben.

Zu beachten ist, dass falls nach dem Stop das Formatprogramm nicht fortgesetzt wird, die Indexregister sowie PM8 nicht regeneriert sind. Wird dies gewuenscht, so ist nach Stop der Fremdbefehl FORM + (J1) + 99 0 10 auszufuehren. In diesem Fall erfolgt nach dem Regenerieren der Indexregister und von PM8 der Ruecksprung ins Hauptprogramm. Bei fehlerfreiem Ablauf erfolgt der Ruecksprung, wenn entweder der Formatstring abgearbeitet ist oder das Ende der Ausgabedaten erreicht ist.

6 . Beispiele :

Im Folgenden wird fuer jede Spezifikation ein Beispiel angegeben.

Spez.	andere Schreibart	(n[1]) bzw (ñ[1])	Ausdruck
4HTEST	-	-	TEST
3PG12.2	3P1G12.2	22345670000048	--234.57 _D -05
F8.6	1F8.6	22345670000048	-.002346
3P2N8.2	-	12345672234567	--234.57--234.57
3PL12.8	3P1L12.8	12345676543210	234.56765432
N3.2	0P1N3.2	1351000	.35
2PN2.0	2P1N2.0	36473391351000	DM 35
B10	1B10	90000199000019
5X	XX3X	-

Mit Hilfe der angefuhrten Spezifikationen koennte etwa folgender Formatstring gebildet werden, der den oben angegebenen Ausdruck zur Folge haette:

(/ 4HTEST / 3PG 12.2 / F8.6 / 3P 2 N 8.2 / 3 PL 12.8 / N3.2 / 2PN2.0 / B 10)

Zum Abschluss sei ein Beispiel angegeben fuer eine Klammerstruktur. Es werde angenommen, dass die Ausgabedaten fortlaufend die Form

(ñ) = 12345673232323

besitzen. Der Formatstring laute :

(/ 29HBEISPIEL FUER KLAMMERSTRUKTUR //
 3 (2 (3PG12.2 , 2X H, 2X F8.6 /) 3P2N8.2 , 3PL12.8 ///)
 4HENDE)

Es erfolgt der Ausdruck :

BEISPIEL FUER KLAMMERSTRUKTUR

234.57_n-30 , .000000
234.57_n-30 , .000000
234.57... 234.57
... 234.56732323

234.57_n-30 , .000000
234.57_n-30 , .000000
234.57... 234.57
... 234.56732323

234.57_n-30 , .000000
234.57_n-30 , .000000
234.57... 234.57
... 234.56732323

ENDE

(Die α -Interpretation der Zahl 23 ist der Punkt .) .