

K. Weise

Physikalisch-Technische Bundesanstalt, Braunschweig

Ein Tischrechenprogramm für Siemens-Prozeßrechner des Systems 300

Ein Tischrechenprogramm für Siemens-Prozeßrechner des Systems 300*)

Zusammenfassung

Mittels des Programmsystems TIRE können die an einer Rechenanlage des Systems 300 angeschlossenen Blattschreiber simultan in einer Art Time-Sharing-Betrieb als programmierbare Tischrechenmaschinen benutzt werden.

Beim Einsatz von Prozeßrechnern im wissenschaftlichen Bereich, z. B. zur Erfassung von Meßdaten kernphysikalischer Experimente, kommt es häufig vor, daß die Experimentatoren kurzfristig und schnell kleinere mathematische Berechnungen durchführen möchten. Eine ALGOL- oder FORTRAN-Programmierung der Probleme ist jedoch oft nicht sinnvoll, da während des tagelang dauernden On-line-Betriebs bei laufenden Messungen meist nicht genügend freier Arbeitsspeicherraum für die Übersetzung und den Ablauf der Programme zur Verfügung steht und eine Vergabe der Programme an eine Großrechenanlage aus Zeit- und Aufwandsgründen oder wegen zu geringer Bedeutung der Probleme nicht in Frage kommt. Andererseits sind die Rechenaufgaben doch so kompliziert, daß kleinere und auch programmierbare Tischrechenmaschinen zu ihrer Lösung nicht geeignet sind. Neuerdings sind auch Tischcomputer auf dem Markt, die programmierbar sind, Standardfunktionen zu berechnen gestatten und bedingte Programmsprünge ausführen können. Diese sind unter den Rechenmaschinen den anstehenden Problemen am besten angepaßt, ihre Anschaffung erfordert jedoch erhebliche zusätzliche Geldmittel. Auch ist nicht einzusehen, warum nicht doch ein vorhandener, wenn auch stark belasteter Prozeßrechner für diese Zwecke benutzt werden kann. Dieser ermöglicht zudem noch den Zugriff zu automatisch erfaßten Daten und den Eingriff in den Prozeß.

Aus diesen Gründen wurde das Programmsystem TIRE entwickelt, welches bis zu einem weiteren Ausbau gestattet, die zur Bedienung zugelassenen Blattschreiber, die am Bedienungs- oder Fernschreibelement einer Siemens-Rechenanlage des Systems 300 angeschlossen sind, simultan in einer Art Time-Sharing-Betrieb und neben einer laufenden Verarbeitung von Prozeßdaten als programmierbare Tischrechenmaschinen zu benutzen. Für die Lösung der auftretenden mathematischen Probleme durch das Programmsystem wurde eine Programmiersprache entworfen, die sich bei der Darstellung von Ausdrücken an FORTRAN anlehnt und sonst so einfach ist, daß sie gerade noch nicht zu schwierige Aufgaben bequem zu handhaben erlaubt. Besondere Anstrengungen wurden unternommen, um den Arbeitsspeicherbedarf möglichst klein zu halten. Bei der fertiggestellten Arbeitsspeicherresidenten Version des Programmsystems benötigt ein zentraler Teil, der ein Vorbereitungs- und Bedienungsprogramm, den interpretativ arbeitenden Übersetzer in Form eines Unterprogramms, die Standardunterprogramme und einen Kellerspeicher für Zwischenergebnisse bei Ausdrücken enthält, fast genau 2 K und jedes der den einzelnen Teilnehmern am Tisch-

*) Mitteilung aus der Physikalisch-Technischen Bundesanstalt

rechenbetrieb zugeordneten Steuerprogramme einen Bereich wählbarer Länge von wenigstens etwa 250 Speicherplätzen.

Die Arbeitsweise des Programmsystems TIRE und die Syntax der Programmiersprache (s. Anhang B) seien an Hand eines einfachen Beispiels (s. Anhang A) erläutert. Nach der Bereitstellung und dem Start von TIRE wird jedem Teilnehmer eine Programmnummer n, ein Arbeitsspeicherbereich (Adresse a bis e) und ein beliebiges Kennzeichen x durch eine Anweisung $TSH=n,a,e,x;$ zugewiesen. Danach und nach Abgabe der Programmdefinitionsanweisung $DEFx;$ kann dieser nun das Tischrechenprogramm in Form einer Folge von durch Semikolon abgeschlossenen Anweisungen entsprechend der Syntax von Hand oder per Lochstreifen eingeben. Eine zusätzliche Korrekturmöglichkeit zu der üblichen durch das Irrungszeichen \neq besteht darin, daß mittels der uneigentlichen Anweisung $^*;$ die vorhergehende gelöscht und danach diese wiederholt werden kann. Die Syntax kennt nur 5 Anweisungstypen: Wertzuweisungen, Ein- und Ausgaben, Marken und bedingte Sprünge. Ein einzelnes Semikolon, gleichzeitig Endanweisung des Tischrechenprogramms, beendet die Eingabe. Durch die Bedienungsanweisung $LSTx;$ kann das geladene Programm aufgelistet und ein Lochstreifen davon erstellt werden. Die Bedienungsanweisung $RUNx;$ löst den Programmablauf aus, der u. U. abgebrochen werden kann durch $HLT=n,x,;$ Die Abarbeitung des Programms erfolgt mit Syntaxkontrolle interpretativ ohne Erstellung eines Maschinencodes. Programmierfehler werden angezeigt und Ausdrücke in Anlehnung an Ref.¹⁾ mit Hilfe rekursiver Unterprogramme berechnet. Nach jeder Anweisung erfolgt ein zyklischer Programmwechsel zum nächsten Teilnehmer. Bei jedem Programmsprung beginnt die Abarbeitung von vorn, jedoch wird das Programm bis zur Zielmarke überlesen. Während des Ablaufs werden Daten in einer standardmäßigen Kolonnenform, bei der auch die Variablennamen erscheinen, ein- oder ausgegeben. Nach Ablauf des Tischrechenprogramms kann ein weiterer Lauf angeschlossen oder auch ein neues Programm eingegeben werden.

Anhang A
Programmbeispiel

Berechnung der Gammafunktion nach der Formel

$$\Gamma(x) \approx \sqrt{2\pi x} \left(\frac{x}{e}\right)^x e^{-1/12x}$$

mit einem relativen Fehler < 3% für $x \geq 1$.

Für $x < 1$ wird $\Gamma(x) = 0$ gesetzt.

```
BRPS=7,2100,TIRE;;BR:STRT=7;;
TSH=8,4200,4600,Q;;DEFQ;;
X:;Y=X-1;Y'1;GAMMA=0;G0'2;1;
GAMMA=SQRT(2*3.141593*X)*X**X*EXP(-X+1/12/X);
5W:*=2;GAMMA;;
/Q;LSTQ;;                                Vorbereitung
                                                Q Teilnehmerkennung
                                                Programmierung
                                                Korrektur durch *;
                                                Liste

X:;                                Eingabe
Y=X-1;                                Wertzuweisung
Y'1;                                Sprung wenn Y >= 0 nach Marke 1
GAMMA=0;                                unbedingter Sprung
G0'2;                                Marke
1;                                GAMMA
GAMMA=SQRT(2*3.141593*X)*X**X*EXP(-X+1/12/X);
2;
GAMMA;                                Ausgabe
;                                Ende
/Q;RUNQ;;                                Ablauf

X      = 2;                                Eingabe
GAMM  = 2.000648;                                Ausgabe
/Q;RUNQ;;
```



```
X      = .5;
GAMM  = 0;
/Q;
ENDW=8;;
ENDW=7;;
```

Anhang B

Syntax der Programmiersprache

1. Programm Das Tischrechenprogramm besteht aus einer Folge von Anweisungen. Ein einzelnes Semikolon markiert als uneigentliche Anweisung das Programmende.

$\langle \text{Programm} \rangle ::= ; \mid \langle \text{Anweisung} \rangle \langle \text{Programm} \rangle$

2. Anweisungen Das letzte Zeichen jeder Anweisung ist ein Semikolon.

$\langle \text{Anweisung} \rangle ::= \langle \text{Marke} \rangle \mid \langle \text{Sprung} \rangle \mid \langle \text{Wertzuweisung} \rangle$
 $\quad \quad \quad \langle \text{Eingabe} \rangle \mid \langle \text{Ausgabe} \rangle$

$\langle \text{Marke} \rangle ::= \langle \text{Ziffernkette} \rangle ; \quad (\text{z. B.: } 1; 037;)$

$\langle \text{Sprung} \rangle ::= \langle \text{Variable} \rangle ' \langle \text{Marke} \rangle \quad (\text{z. B.: } Y'5; GO'10;)$

Ist der Wert der Variablen nicht negativ, so wird nach der angegebenen Marke gesprungen.
Sonderfall: Die Variable GO wird nicht abgefragt, sondern ein unbedingter Sprung ausgeführt.

$\langle \text{Wertzuweisung} \rangle ::= \langle \text{Variable} \rangle = \langle \text{Ausdruck} \rangle;$

(z. B.: A=B**(-2/3);Z=X-SIN(Y);)

$\langle \text{Eingabe} \rangle ::= \langle \text{Variable} \rangle : ; \quad (\text{z. B.: } PETER::)$

Der Wert der Variablen wird nach Ausgabe ihres Namens über Blattschreiber angefordert und ist in der Form $\langle \text{reelle Zahl} \rangle$ (s.5.) einzugeben (z. B.: PETE = -2.5\$-3;).

$\langle \text{Ausgabe} \rangle ::= \langle \text{Variable} \rangle ; \quad (\text{z. B.: } OTTO;)$

Der Name und der Wert der Variablen werden über Blattschreiber in der Form $\langle \text{reelle Zahl} \rangle$ (s.5.) ausgegeben (z. B.: OTTO = -2.500000\$-13;).

$\langle \text{Variable} \rangle ::= \langle \text{Name} \rangle$

Alle Variablen und auch alle anderen Zahlengrößen sind vom Typ Real.

3. Zeichen und Zeichenketten

$\langle \text{Ziffernkette} \rangle ::= \langle \text{Ziffer} \rangle \mid \langle \text{Ziffernkette} \rangle \langle \text{Ziffer} \rangle$
 $\langle \text{Name} \rangle ::= \langle \text{Buchstabe} \rangle \mid \langle \text{Name} \rangle \langle \text{Ziffer} \rangle \mid$
 $\quad \quad \quad \langle \text{Name} \rangle \langle \text{Buchstabe} \rangle \quad (\text{z. B.: } BERLIN\ 1)$

Nur die maximal 4 ersten Zeichen eines Namens oder der Ziffernkette bei einer Marke sind von Bedeutung, die übrigen redundant.

$\langle \text{Ziffer} \rangle ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

$\langle \text{Buchstabe} \rangle ::= A \mid B \mid C \mid D \mid E \mid F \mid G \mid H \mid I \mid J \mid K \mid L \mid M \mid N \mid O \mid P \mid Q \mid R \mid S \mid T \mid$
 $\quad \quad \quad U \mid V \mid W \mid X \mid Y \mid Z$

$\langle \pm \rangle ::= + \mid -$

$\langle \ast \rangle ::= * \mid /$

Zeichen „Zwischenraum“ können beliebig in das Programm eingestreut werden. Sie werden überlesen.

4. Ausdrücke Schreibweise wie in FORTRAN

$\langle \text{Ausdruck} \rangle ::= \langle \text{Glied} \rangle \mid \langle \pm \rangle \langle \text{Glied} \rangle \mid$
 $\quad \quad \quad \langle \text{Ausdruck} \rangle \langle \pm \rangle \langle \text{Glied} \rangle$
 $\langle \text{Glied} \rangle ::= \langle \text{Faktor} \rangle \mid \langle \text{Glied} \rangle \langle \ast \rangle \langle \text{Faktor} \rangle$
 $\langle \text{Faktor} \rangle ::= \langle \text{Primausdruck} \rangle \mid \langle \text{Faktor} \rangle \ast \ast \langle \text{Primausdruck} \rangle$
 $\langle \text{Primausdruck} \rangle ::= \langle \text{Zahl} \rangle \mid \langle \text{Variable} \rangle \mid (\langle \text{Ausdruck} \rangle) \mid$
 $\quad \quad \quad \langle \text{Funktionsaufruf} \rangle$
 $\langle \text{Funktionsaufruf} \rangle ::= \langle \text{Name} \rangle (\langle \text{Ausdruck} \rangle)$
Bei Funktionsaufrufen sind nur die Namen EXP, SIN, COS, ABS, LN, SQRT, ARCTAN (hier genügt ARCT) erlaubt, d. h. nur die der Standardfunktionen.

5. Zahlen Alle Zahlen werden als Gleitpunktzahlen behandelt. Sie müssen im erlaubten Zahlenbereich des Systems 300 liegen.

$\langle \text{Zahl} \rangle ::= \langle \text{Mantisse} \rangle | \langle \text{Mantisse} \rangle \langle \text{Exponent} \rangle | \langle \text{Exponent} \rangle$

$\langle \text{Mantisse} \rangle ::= \langle \text{Zifternkette} \rangle | \langle \text{Zifternkette} \rangle . |$

$. \langle \text{Zifternkette} \rangle | \langle \text{Zifternkette} \rangle . \langle \text{Zifternkette} \rangle$

$\langle \text{Exponent} \rangle ::= \$ \langle \text{Zifternkette} \rangle | \$ \langle \pm \rangle \langle \text{Zifternkette} \rangle$

$\langle \text{reelle Zahl} \rangle ::= \langle \text{Zahl} \rangle ; | \langle \pm \rangle \langle \text{Zahl} \rangle ;$

Literatur

1) F.E.J. Kruseman Aretz, Elektronische Datenverarbeitung 6 (1964) 233