

Programmiersystem

Verwendungszweck

Das Programmiersystem bildet in unterschiedlichen Zusammensetzungen mit dem Betriebssystem BS 3 die Programmausstattung des Teilnehmer-Rechensystems TR 440. Durch umfangreiche Testhilfen ist es, speziell im Dialogbetrieb, für alle Anwendungsfälle verwendbar.

Besondere Merkmale

Als Programmiersystem werden alle Programme (Operatoren genannt) und Prozeduren zusammengefaßt, die in Dateien der öffentlichen Datenbasis (Systembibliothek) gehalten werden.

Um dem Benutzer ein einheitliches und bequemes Steuermittel für den Bearbeitungsablauf in die Hand zu geben, wurde die **TR 440-Kommandosprache** geschaffen.

Der Übersetzungsvorgang aus einer Programmiersprache in eine der Rechenanlage verständlichen Form erfolgt zweistufig über Quelle → Montageobjekt → Operator. (s. u.)

In Quellenprogrammen aufgerufene Prozeduren (des Systems oder des Benutzers) werden erst bei dem Montagevorgang angeschlossen.

Prozeduren, die in verschiedenen Sprachen formuliert sind, können gegenseitig aufgerufen werden. Auf diese Weise können die Vorteile der verschiedenen Programmiersprachen auch von anderen Sprachen ausgenutzt werden.

Bei irregulären Abbrüchen der Auftragsbearbeitung werden **quellenbezogene Dumps** erstellt, d. h. die in den Quellen angegebenen Namen von Variablen und deren Inhalt werden in lesbarer Form ausgegeben.

Der Programmablauf kann verfolgt werden, indem z. B. die Protokollierung aller auftretenden Sprungbefehle oder Zuweisungen gewünscht wird. Diese **Ablaufüberwachung** (tracing) ist auch in Programmen aus höheren Sprachen möglich, ohne die Quelle verändern zu müssen.

Auf Wunsch des Benutzers werden beim Übersetzungsvorgang **dynamische Kontrollen** eincompiliert, mit deren

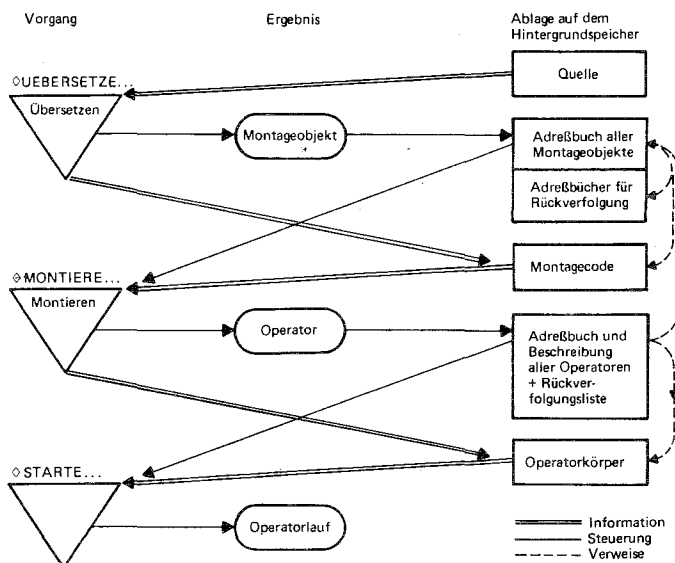
Hilfe auch statisch und syntaktisch nicht erkennbare Fehler lokalisiert werden können (Einhaltung von Indexgrenzen o. ä.). Genaueres über die Testhilfen, insbesondere im Dialogmodus, kann einem gesonderten Informationsblatt entnommen werden.

Sprachübersetzer

Welche der hier aufgeführten Übersetzer Bestandteil des Programmiersystems sind, liegt am Einsatz des Rechensystems und an den Wünschen des Rechenzentrums.

Die TELEFUNKEN-Assemblersprache **TAS** ist eine formatfreie Sprache mit symbolischer Adressierung und 240 Operationscodes.

Der **FORTRAN**-Compiler übersetzt Programme im ASA-Standard oder in FORTRAN IV H. Der Sprachumfang wurde durch Stringmanipulationen erweitert, die Textverarbeitung ermöglichen.



Übersetzungsvorgang im Programmiersystem

Der **ALGOL**-Compiler verarbeitet den Sprachumfang von ALGOL 60. Als Erweiterung wurde ein fortranähnliches COMMON-Konzept realisiert. Es steht die ALCOR-, ISO- und Knuth-EA-Prozeduren zur Verfügung.

Der **COBOL**-Compiler verarbeitet den Sprachumfang des ANSI-Standard COBOL 68.

GPSS (General Purpose Simulation System) ist eine Sprache für Probleme der diskreten Systemsimulation.

Der übliche **RPG**-Sprachumfang wurde im Hinblick auf den Tabellenaufbau wesentlich erweitert.

Der **BASIC**-Interpreter ist voll gesprächsfähig und umfaßt den üblicherweise installierten Sprachumfang.

Der Sprachumfang des **PL/I**-Compilers entspricht dem derzeitigen Stand der internationalen Normung.

TR 440-Kommandosprache

Mit der Kommandosprache läßt sich die gesamte Externsteuerung des Rechensystems formulieren. Ein Kommando wird mit einem Fluchtsymbol (dargestellt mit ◇) eingeleitet, um es von den übrigen Eingabedaten unterscheiden zu können.

Das Grundelement der Kommandosprache ist das **Tätigkeitskommando**, mit dem eine bestimmte Leistung vom Programmiersystem verlangt wird. Es beginnt mit einem Tätigkeitsnamen, auf den Spezifikationen folgen, z. B.:

◇ UEBERSETZE, SPRACHE = FTN, QUELLE = TEXT, ...

Für jede Tätigkeit ist eine Menge von **Spezifikationen** definiert (im Beispiel SPRACHE, QUELLE usw.).

Hält man eine vorgegebene Reihenfolge ein, so können die Spezifikationsnamen entfallen. Auch das Abkürzen der Namen ist möglich, solange Eindeutigkeit gewährleistet ist, z. B.:

◇ UEB., FTN., QU. = TEXT, ...

Zu den einzelnen Spezifikationen, die in obligate und optionale unterteilt werden, existiert eine globale Voreinstellung, wenn keine Aussagen dazu gemacht werden.

Der vorhandene Kommandovorrat läßt sich über das DEFINIERE-Kommando oder über eine Prozedurvereinbarung erweitern, ohne daß dabei bestehende Kommandos versehentlich gelöscht werden können.

Um eine vorgegebene Kommandofolge zu verlassen, gibt es zwei spezielle Kommandos, die den Auftrag nach einem aufgetretenen Fehler abbrechen (FEHLERHALT) oder in Abhängigkeit von auftragspezifischen Variablen auf nachfolgende Kommandos springen (SPRINGE).

Operatoren

Ein Operator im Sinne der Softwareorganisation des TR 440 ist jedes Programm, das folgenden Bedingungen genügt:

— alle Beziehungen zum Betriebssystem werden über einen Abwicklerprozeß hergestellt (etwa als benutzer-spezifisches Betriebssystem zu bezeichnen)

— im Abwickler existiert eine Beschreibung dieses Programms (Startadresse, Größe usw.). Aufgrund dieser Beschreibung kann der Abwickler einen Operator starten.

Ein Benutzerauftrag besteht aus der eigentlichen Information, die von der Rechenanlage zu verarbeiten ist (Quelle, Daten) und Angaben darüber, in welcher Form das geschehen soll. Diese Steuerinformation in Form von Kommandos wird von einem speziellen Operator, dem **Entschlüßler**, ausgewertet. Der Entschlüßler veranlaßt daraufhin den Start der geforderten Operatoren.

Auch das Programm des Benutzers wird zu einem Operatorlauf in der Kette der Bearbeitungen (job steps).

Die Beschreibung und die Erzeugung des Programms übernimmt ein spezieller Operator, der **Montierer**. Zu der Beschreibung gehören auf Wunsch des Benutzers auch

Referenzlisten, die es den zahlreichen **Analyseoperatoren** (Rückverfolger, quellenbezogene Dumps u. a.) erlauben, im Fehlerfall des Programmablaufs Information mit den Bezeichnungen in der Quelle des Benutzers auszugeben und die Fehlerstelle zu lokalisieren.

Übersetzungsvorgang

Eine der wichtigsten Leistungen des Programmiersystems ist das Umsetzen einer Quelle in die gewünschte Rechnerleistung (vergl. das Bild auf der Vorderseite).

Die **Quelle**, die entweder in einer Datei (file) oder in der Form von Eingabedaten auf einem Hintergrundspeicher liegt, wird zunächst von einem Sprachübersetzer verarbeitet. Dieser erzeugt daraus ein **Montageobjekt**, das aus dem entsprechenden Montagecode und aus Adreßbüchern für die Rückverfolgung im Fehlerfall besteht, falls Dumpfähigkeit verlangt wird. Der Montierer sorgt für die Verknüpfung und günstige Anordnung von einem oder mehreren Montageobjekten in dem von ihm daraus erstellten, lauffähigen **Operator**.

Die Zweistufigkeit der Übersetzung kann u. a. dazu verwendet werden, Prozeduren aus verschiedenen Sprachen gemeinsam zu verwenden.

Dabei ist der gegenseitige Aufruf in den Sprachen ALGOL 60, FORTRAN und TAS realisiert. Aus TAS heraus können auch COBOL-Routinen, aus COBOL Routinen der restlichen genannten Sprachen aufgerufen werden.

Dialogbetrieb

Im Gespräch von einer Benutzerstation stehen bis auf unwesentliche Einschränkungen alle Dienstleistungen des Programmiersystems zur Verfügung. Im Dialogmodus können in eine bereits eingegebene Kommandofolge Kommandos oder Anweisungen eingestreut werden (vergl. das Informationsblatt „Dialogbetrieb, Testhilfen“, N31.B0.15).

Datenorganisation

Zur Speicherung von Datenbeständen stehen dem Benutzer die Speichermedien Trommel, Fest- oder Wechselplatte und Magnetband zur Verfügung. Die Information wird in Form von Dateien (files) gehalten. Nach der Art des Zugriffs auf die Sätze einer Datei werden vier Dateitypen unterschieden:

SEQ Sequentieller Zugriff gemäß der physikalischen Reihenfolge

RAN Randomzugriff über Satznummern, die sequentiell organisiert werden

RAM Randomzugriff über Satzmarken, die index-sequentiell angeordnet sind.

RAS Randomzugriff über den Satzschlüssel; die Sätze sind dem Satzschlüssel entsprechend in aufsteigender Reihenfolge angeordnet.

Ein fünfter Dateityp — **PHYS** — gestattet den direkten Zugriff auf Blöcke (zu 128 Ganzworten)

Bis zu 255 Dateien können für einen Auftrag existieren.

Die Daten werden in verschiedenen Bibliotheken (data base) gehalten, die sich durch ihre Lebensdauer und Zugriffseigenschaften unterscheiden.

In der **öffentlichen Datenbasis** (Systembibliothek) liegt der Programmbestand des Programmiersystems.

Eine **Standard-Datenbasis** (Auftragsbibliothek) steht jedem Benutzerauftrag für die Lebensdauer dieses Auftrags zur Verfügung.

Die **Langfristige Datenhaltung** (Benutzerbibliothek) bietet die Möglichkeit, Dateien permanent zu speichern (unabhängig von der Lebensdauer des Systems oder des Benutzerauftrags).