

Ein Abwicklerprozeß verwaltet die einzelnen Programme (Operatoren), die der Benutzer mit Kommandos anspricht und die ihm die gewünschte Rechenleistung erbringen. Es entsteht mit Hilfe des SAV ein Wechselgespräch zwischen dem Benutzer und den von ihm angesprochenen Operatoren oder auch direkt mit dem Abwicklerprozeß. Die Information wird dabei je nach Umfang im Kernspeicher oder über den Plattenspeicher ausgetauscht. Zusätzlich kann der Benutzer auf seine Datenbestände in der Langfristigen Datenhaltung zugreifen.

Voraussetzungen für Testhilfen

Die im folgenden erläuterten Testhilfen können nur in Anspruch genommen werden, wenn schon im UEBERSETZE-Kommando die Gesprächsfähigkeit (mit der Spezifikation VARIANTE), die Ablaufüberwachung (TRACE), die dynamischen Kontrollen (DYNKON) oder die Verwaltung von Kontrollereignissen (KE) gewünscht werden.

Der Compiler baut daraufhin die entsprechenden Prozeduren ein und der Montierer legt Referenzlisten für die zahlreichen Analyseoperatoren an.

Im STARTE-Kommando gibt der Benutzer an, welche Dumps er im Fehlerfall wo ausgeben (DUMP), welche Kontrollereignisse er aktivieren möchte (AKTIV) und ob eine Überwachersteuerung erfolgen soll (UEBWS).

Testmöglichkeiten für alle Aufträge

Der Benutzer kann seinen Auftrag auch im Abschnittsmodus (batch) stellen. Er hat dabei auf den Gang der Bearbeitung keinen Einfluß mehr, kann aber ebenso wie im Gesprächsmodus die folgenden Testhilfen verwenden:

Quellenbezogene Dumps drucken bei irregulären Abbrüchen (im Dialog auch auf Wunsch) der Bearbeitung die in den Quellen angegebenen Namen von Variablen und deren Inhalt in lesbarer Form aus, so daß Referenzlisten nicht benutzt werden müssen.

Es kann u. a. der Ausdruck aller Variablen mit einzelnen Ausnahmen (z. B. große Felder) oder nur der Variablen, die an der aktuellen Aufrufverschachtelung beteiligt sind, oder der Druck explizit angegebener Variablen verlangt werden.

Wurden Prozeduren verschiedener Quellsprachen zusammengeschlossen, so kann für jede Sprache der Dump entsprechend quellenbezogen gewünscht werden. Beispiel:

```
... , DUMP = F - KONSOL (A,B), ...
```

Die FORTRAN-Variablen (es waren im Beispiel Felder) sollen auf die Konsole (Benutzerstation) ausgegeben werden. Ein Element von A würde dann etwa so aussehen:

```
    FELD A  
    A (1)  
    0.50000000000 E 001
```

Der **Rückverfolger** würde vor den Dumps noch quellenbezogen die aktuelle Aufrufverschachtelung bis zur Quellzeile des Hauptprogramms und eine mögliche Fehlerursache ausdrucken.

Die **Ablaufüberwachung** (tracing) gibt die Möglichkeit, z. B. die Protokollierung aller Sprünge oder aller Anweisungen zu veranlassen. Als Beispiel soll die Ablaufüberwachung mit einer Anweisung (s. dort) für Zuweisungen im Gesprächsmodus eingeschaltet werden. ...◇: KTRACEIN (ASSIGN)

Drei Zeilen sollen das Ergebnis demonstrieren:

```
** IN ZEILE 100 ZUW.: I      = 1  
** IN ZEILE 110 ZUW.: A (2) =  0.600000000000 E+001  
** IN ZEILE 120 ZUW.: B (2) =  0.360000000000 E+002
```

Eine weitere Möglichkeit der Ablaufüberwachung, die Rechenzeit und Druckseiten spart, bietet das **Backtracing**. Hierbei erfolgt im Fehlerfall oder auf explizite Anforderung ein Ausstülpen des Backtrace-puffers. Abhängig von der Art der Information werden die max. 50 letzten Schnitte ausgegeben. Es werden dabei alle vereinbarten Überwachungen berücksichtigt.

Dynamische Kontrollen schließlich werden auf Wunsch des Benutzers beim Übersetzungsvorgang eincompiliert und beziehen sich auf

- Einhaltung von Feldgrenzen,
- Verträglichkeit von formalen und aktuellen Parametern,
- Zulässigkeit von Schleifenparametern.

Damit lassen sich statisch und syntaktisch nicht erkennbare Fehler lokalisieren.

Erweiterte Testmöglichkeiten

Das Austesten eines Programms wird gerade im Gesprächsmodus durch die Möglichkeit, auf Ergebnisse sofort zu reagieren, vereinfacht und beschleunigt. Ein geschäftsfähig montiertes Programm kann jederzeit während seines Laufs angehalten werden.

Im UEBERSETZE-Kommando werden sogenannte **Kontrollereignisse** definiert, indem den Nummern von Quellzeilen Namen zugeordnet werden.

Ist ein Kontrollereignis aktiv (Angabe im STARTE-Kommando), so hält das Programm an der entsprechenden Stelle an und meldet den Namen des Ereignisses an der Benutzerstation. Beispiel:

```
... , KE = 20-FTN1'90-FTN2, ...  
... , AKTIV = KEINE (FTN1), ...
```

Die Quellzeilen 20 und 90 werden als Kontrollereignisse definiert, FTN1 wird beim Programmstart aktiviert:

```
STDHP*KE=FTN1 ◇: KEAKTIV (FTN2)
```

Auf ein Kontrollereignis kann mit der Eingabe von Kommandos oder Anweisungen reagiert werden. Im Beispiel wurde das Kontrollereignis FTN2 mit einer Anweisung aktiviert. (STDHP ist der Name des Programm- laufs, falls der Benutzer keinen eigenen vergeben hat).

Anweisungen werden immer einzeln und nur auf Anforderung verarbeitet. Der Kommandoentschlüssler versteht Anweisungen zum

- Fortsetzen und Beenden des Programmlaufs,
- Aktivieren und Passivieren von Kontrollereignissen, Außerdem kann auf die weitere Bearbeitung Einfluß genommen werden durch:
 - Bringen oder Umsetzen der Werte einzelner Variablen, die frei aus dem Programm wählbar sind,
 - Dump der Variablen,
 - Löschen aller noch anstehenden Kommandos oder
 - erneutes Anhalten bei Programmende.

Der Benutzer kann eine vorrangige Ausführung von Kommandos dadurch erreichen, daß er Kommandos einschiebt, wenn sich der Entschlüssler nach einer fehlerhaften Kommandoausführung meldet. Auf diese Weise kann beispielsweise eine fehlerhafte Montage wiederholt werden, bevor das schon eingegebene STARTE-Kommando wirksam wird.

Texthaltung

Arbeitet der Benutzer im Gespräch mit umfangreichen Quellen oder Datenmengen, so ist es unbequem, diese von der Benutzerstation aus einzugeben. Vielmehr ist in diesem Fall der Gebrauch der langfristigen Datenhaltung LFD zu empfehlen.

Der Benutzer richtet sich dabei auf einem Hintergrundspeicher Daten ein, die er in einem Abschnitt füllt und deren Lebensdauer er selbst bestimmt (Benutzerbibliothek, unabhängig von der Lebensdauer des Systems oder eines Benutzerauftrags).

Eng im Zusammenhang mit diesen Dateien steht die Texthaltung, wenngleich diese auch nur für die Lebensdauer eines Dialogs benutzt werden kann.

Mit ihrer Hilfe lassen sich Quellen (oder strukturierte Daten) in Dateien eintragen oder löschen. Einzelne Zeilen oder Zeichen können korrigiert, gelöscht oder auch vertauscht werden. Ferner kann der Benutzer die Dateien (auch Kapitel genannt) mischen, zusammenfügen oder Teile davon kopieren, sowie sich über ihren Inhalt informieren.

Die Texthaltung erübrigt geschäftsfähige Compiler, da einzelne Prozeduren korrigiert und erneut als Ganzes übersetzt werden können. Diese Handhabung ist einem interpretativen System in Hinblick auf Rechenzeit- und Speicherbedarf mindestens ebenbürtig, wenn nicht überlegen.

Spezielle Testhilfe-Kommandos

Neben den bisher beschriebenen Möglichkeiten erlauben die Kommandos **THSETZE** und **DTESTE** weitere Testhilfen.

Mit dem **THSETZE**-Kommando kann man Testhilfeangaben für nachfolgende Programmläufe hinterlegen, insbesondere für nicht direkt oder mit dem **STARTE**-Kommando gestartete Läufe.

Außerdem bietet sich die Möglichkeit, die hinterlegten Testhilfeangaben mit dem **STARTE**-Kommando (**LAUF**) abzurufen.

Das Kommando **DTESTE** überprüft Dateien auf Übereinstimmung von Verwaltungsinformation und Dateistruktur sowie auf Lesbarkeit der Information.