

3D-8

デバッガでの 多言語共通フルスクリーンデバッグ機能

田川 豊 , 磯谷 利夫

((株) 日立製作所 ソフトウェア工場)

1. はじめに

近年のソフト産業においては、経験の浅いプログラマの急速な増加や、VDTの普及にともない、誰でも簡単に使用できる対話型デバッガが求められてきた。

この要求に対し、日立では多言語共通の対話型デバッガを開発した。本稿では、多言語共通デバッガの機能のなかで、ソースプログラムの内容を画面表示し、実際のプログラムの動きを見ながらデバッグできるフルスクリーンデバッグ機能を中心に述べる。

2. 多言語共通デバッガの構造

対話型デバッガの機能は、デバッグ対象プログラムの実行を監視しながら、ユーザがVDTから入力したサブコマンド(注1)を解析・実行し、デバッグ情報を提供することである。対話型デバッガの基本的な機能は、次の四つである。

- (1) 変数の参照と変数への設定
- (2) 実行中断
- (3) 実行順序の変更
- (4) トレース

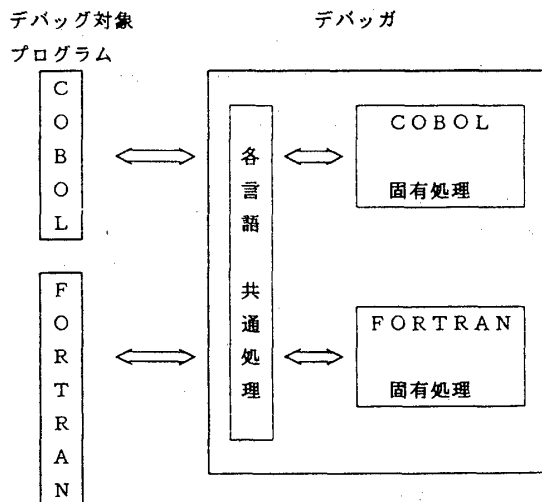


図1 多言語共通デバッガの構造

ここで、デバッガの構造は、大きく分けて、言語に依存する部分と依存しない部分の二つに分けることができる。この構造を、今回実現したCOBOL、FORTRAN対応の多言語共通デバッガを例にとって、図1に示す。

各言語により、変数の文法は異なるので、各コンパイラからの変数に関するデバッグ情報を、同一形式にすることはできない。このため、(1)についてはデバッガは言語を意識し、言語固有の処理を行わなければならない。これが言語に依存する部分である。

しかし、(2)、(3)、(4)は、デバッガ内で閉じた処理であり、言語を意識せずに共通化できる。これが言語に依存しない部分である。

(注1): ユーザのデバッガに対する指示をサブコマンドと言う。

3. フルスクリーンデバッグ機能

(1) フルスクリーンの必然性

従来の大型システムにおける対話型デバッガは、ラインモード(注2)であり、次の欠点がある。

- ・ユーザは常に最新リストをプリンタに出力し、そのリストとVDTを見比べながらデバッグしなければならない。

- ・初心者にとっては、プログラムの動きを予測することが難しく、デバッグに労力を要する。

ラインモードのこれらの欠点を解決するのが、フルスクリーンデバッグ機能の目的である。

(注2): デバッグ情報が行単位で表示される方式であり、プログラムの動きをソースレベルで見ることができない。

(2) 機能

図2に、COBOLのフルスクリーンデバッグ機能の画面レイアウトを示す。

```

000170 READ TBL1 AT END MOVE '1' TO EOF.
000180 CALL 'SUB' USING A-DATA R-DATA.
000190 DISPLAY 'R-DATA=' R-DATA.
000200 MOVE A-DATA TO R-DATA.
000210 IF EOF='1' GO TO ENDPOS.

NAME A-DATA
VALUE 500

コマンド [ DISPLAY VALUE A-DATA ]
    
```

図2 画面レイアウト

ソース表示部に、ソースプログラムを表示する。現在実行中の行は、輝度を変えて表す。図2では、文番号190が実行行である。この輝度の異なる行が、次々と移り変わることによって、プログラムの動きを知ることができる。

データ表示部には、デバッグ情報を表示する。図2の最終行のコマンド投入欄に、「DISPLAY VALUE A-DATA」（A-DATAの値を表示せよ）と入力すると、データ表示部に、その内要が表示される。

(3) 実現方式

フルスクリーンデバッグ機能を実現させるためには、基本的に次の二つの情報が必要である。

- (a) ソースプログラム情報
- (b) 現在実行中の行番号

ここで (a) と (b) のコンパイラとデバッガとのインターフェースが同一であれば、一つ一つの言語を意識する必要がない。

そこで、COBOLとFORTRANの各コンパイラからの、(a)、(b)のインターフェースを同一にし、図1に示す「各言語共通処理」部にフルスクリーンデバッグ機能を組み込むことによって、多言語共通フルスクリーンデバッグ機能を実現した。

(4) 特長

- (a) 多言語混在プログラムのデバッグ可能

COBOLとFORTRANが混在するプログラムでも、プログラム名で修飾することにより、任意の点で、デバッガの機能（実行中断など）が使用できる。図3に、実行中断機能の例を示す。

この例では、COBOLプログラム(A)の文番号200と、FORTRANプログラム(B)の文番号500で、中断点を設定する。COBOLプログラムを開始すると、COBOLプログラムの文番号200で中断し、さらに続行するとFORTRA

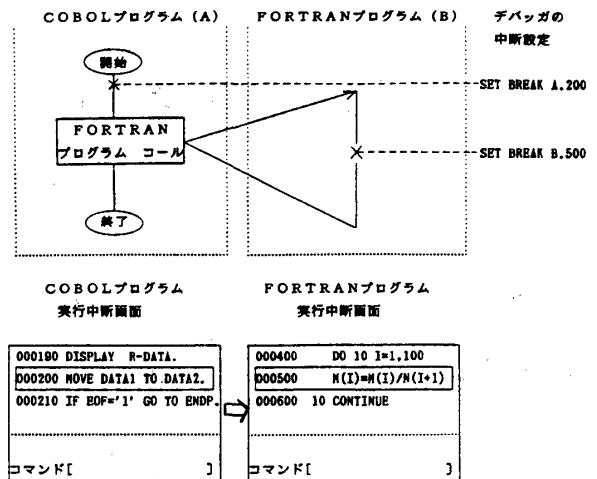


図3. 多言語混在プログラムの実行例

Nプログラムの文番号500で中断する。

またこの例でも分かるように、現在実行中のソースプログラムが、COBOL、FORTRANにかかわらず、同一画面フォーマットで表示できる。

(b) 表示画面操作

フルスクリーンデバッグ機能では、ソースプログラムを見易くするため、表1の表示画面レイアウト変更機能を導入した。

表1 レイアウト変更機能

機能	内容
スクロール	ソース表示部とデータ表示部を、別々に上下左右スクロールできる。
表示行数の変更	ソース表示部とデータ表示部の行数を、自由に変更できる
画面の移動	任意の行を画面に位置づけ、表示できる。

(c) 同一画面内での入出力命令

COBOLのコンソール入出力命令(DISPLAY命令やACCEPT命令)の入出力を、図2で示したフルスクリーン機能画面内で実行できる。DISPLAY命令は、データ表示域に表示し、ACCEPT命令は、コマンド投入欄から入力できる。

4. おわりに

多言語共通デバッガを、COBOLとFORTRANで実現できた。

今後、デバッガからエディタやコンパイラを呼べるようにするなど、さらに使い易いフルスクリーンデバッグ機能を開発していきたいと考えている。