5ZB-07　BASICインタプリタによるデバック支援環境「LQB」の開発

長谷川拓矢　岩澤京子
拓殖大学工学部情報工学科

1 はじめに

1.1 目的

プログラムを学ぶことは難しいものである．これから学習をはじめる初心者に，少しでもプログラミングを習得しやすくできる環境を提供できないかを考えた．

本研究では，デバックに重点を置き，デバックに必要な情報をシステムで扱うことのできるインタプリタを作製した．デバックの方法や着眼点を学んでもらうことを目標としている．

1.2 方針

初心者の対象言語として，次の点から BASIC を対象とする言語に選んだ．

(1) 習得が（他の言語に比べて）容易である．
(2) 一般の手続き型言語への移行ができる．

さらに，構造化プログラミングに対応していることから，マイクロソフト社の「QuickBASIC4.5」[2] の言語仕様を元にしたインタプリタ部をコアとして「LittleQuickBASIC」（以後 LQB）と名づけるシステムを開発した．

LQB は PC-UNIX 上で動作し，ユーザーインターフェイスに GTK+ [1] を用いている．

2 LQB 機能

2.1 LQB 言語仕様

インタプリタ部は上記の通り，QuickBASIC を参考にした．しかし，上記方針のため以下の通りの変更を加える．

(1) 行番号制度の廃止．古い技術である行番号制度のプログラミングを学ばせないで，構造化プログラミングを学ばせるための処置．
(2) 不要な機能の削除．学習意欲を低下させないため，できる限り初心者にとって不要である機能を削除する．これにより「メタコマンド」「グローバル変数」を除外する．

また，実行文・組込関数は必要なものから順次組み込んでいく．現在では 12 の実行文と 22 の組込関数を搭載している．

2.2 自動可視化モード

初心者が解消困難なバグに「実行時エラー」がある．これは構文は合っているが，計算式の記述ミスや，ループの条件式のミスなどで，予想外の動作をプログラムが行うというものである．

そこで，「このモードでソースファイルを実行するだけで，ソースをステップ実行し，そのとき使用者に必要と思われる情報を表示する」デバック用モードを開発した．このモードのことを「自動可視化モード」と呼ぶ．

バグ発見モードの実行は，起動時に「-search」スイッチで実行するか，ファイル名入力ダイアログで指定する．

バグ発見モードは，少なくともループの終了判定の誤りと計算式の異常は発見できるように，以下の６つの情報を表示する．これらは主に制御の流れを追えることを目的としている．

(1) 現在実行中のソースファイル上の位置
(2) 論理式の真偽
(3) ループ実行回数
(4) 分岐命令 (SELECT) の条件結果
(5) その時点での全変数とその値（配列は情報が膨大になってしまうので除く）
(6) 関数の戻り値と，呼び出し前・後の引数の値

以下に，バグ発見モードでプログラムを実行した時の出力画面図１に示す．

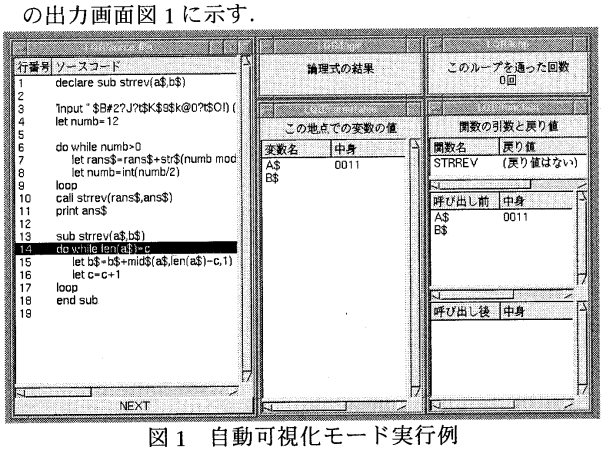

図１　自動可視化モード実行例

2.3 デバック指示文モード

バグ発見モードの機能では，「最後の部分だけ確かめたいのに途中を飛ばすことができない」「変数の変化の過程を見たいのに，情報をとっておくことができない」などの不満が出てくる．さらに，バグ発見モードは他の開発環境にはないので，他の開発環境への移

Development of programing environment for debugging by Basic interpreter
Takuya Hasegawa,Kyouko Iwasawa
Computer Science Department,Takushoku University

行を妨げることになってしまう．そこで，指示文の形でソースファイルに必要な情報を得るための命令を記述し，プログラムの実行に従ってファイルに出力する．この形式により，使用者は実行後でもプログラムの解析をしてバグの発見ができる．そのため，比較的大規模なプログラムのデバックに適している．

このモードは下記の 7 つの指示文（以後「デバック指示文」）を使用し，ユーザが必要とする情報だけを得ることのできるようにしたモードである．このモードを「デバック指示文モード」と呼ぶ．

このモードは自動可視化モードに比べ，「ソースに実行文を記述する」という，中級者以上の人が実行時エラーを修正するときに行う行動に近い動作を用いるようにしている．そのため，このモードに慣れれば，以後は他の開発環境でも，その開発環境のコンソール出力命令やファイル出力命令などを使い，実行時エラーのチェックができるようになるわけである．

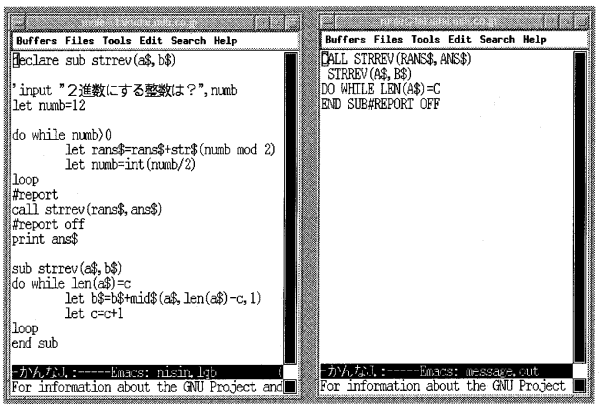

図 2　自動可視化モード実行例

デバック指示文は次のものを用意している．

(1) #FUNCTEST．関数のテスト．特定の引数を与えた場合の引数と戻り値を出力させる．

(2) #VALTEST．変数・式のテスト．その時点での変数・式の内容を計算し，ファイル出力する．

(3) #NAMET．全変数の出力．その時点での全変数（バグ発見と異なり配列も含む）の値をファイル出力する．

(4) #REPORT．実行地点の出力．解析したソースのトークンをファイル出力する．出力するファイルが読みづらく・長くなるのが欠点．

(5) #WATCH．変数変化の監視．変数の変化を監視し，指定した変数の値が変わるときにその変化をファイルに出力する．

(6) #WAIT．ブレークポイント．ブレークポイントを入れる．実行が中断される．

(7) #ALLWAIT：全文ブレークポイント．全実行文の後に入力待ちが行われる．

3　LQB システム構成

LQB の処理は図 3 の通りである．このとき，「自動可視化モード」と「デバック指示文モード」は指定されているときにだけ行われる．

処理は構文構文部がソースファイルをトークンに切り分け，それを実行文処理部が受け取り指定された処理を行う．この動作をファイルが終わるまで繰り返す．

実行される情報のうち，2 つのデバック機能にて扱う情報は，それぞれの処理部に渡される．自動可視化モードは 2.2 の情報を受け取り，ウィンドウに表示する．

また，デバック指示文モードは 2.3 に書かれた 7 つの指示文からのデータを，構文解析部か実行処理部から受け取り，ファイルに出力する．

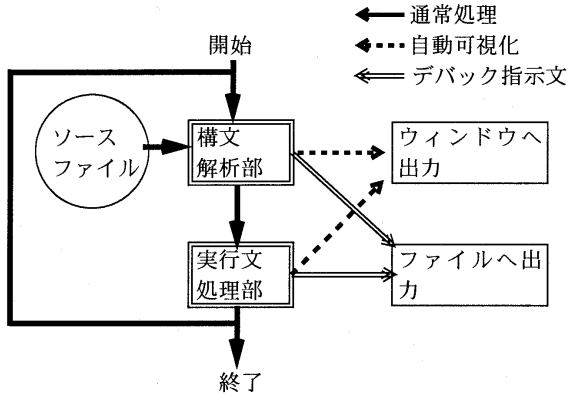

図 3　処理の流れ

4　終わりに

初心者のためのデバック機能として，「自動可視化モード」「デバック指示文モード」の 2 つのモードのあるデバック支援環境を開発した．

仕様としては初心者に十分だと考えるが，BASIC のグラフィック関連の実行文の対応と，実際に初心者に使用してもらった時の評価が今後の課題である．

参考文献

(1) 竹田英二：ＧＴＫ＋ではじめるＸプログラミング，技術評論社

(2) 佐々木整：わかりやすい QuickBASIC4.5，秀和システム

(3) LeendertAmmeraa：C で学ぶデータ構造とプログラミング，オーム社

(4) 中田青男：コンパイラ，オーム社