

TITULO:

Execução Controlada de Programas

AUTORES:

José Roque Voltolini da Silva * +
 Deoni Luiz Segalin ** +
 Renata Vieira *** +
 Paulo A. de Azevedo ****

ENTIDADE:

Universidade Federal do Rio Grande do Sul
 Curso de Pós-Graduação em Ciência da Computação
 Av. Osvaldo Aranha, 99
 Porto Alegre - RS
 Fone: (0512) 21-2161

RESUMO:

Este trabalho descreve um método para execução controlada de programas escritos em Pascal, objetivando auxiliar o programador no processo de desenvolvimento e depuração de programas.

A execução do programa pode ser normal ou reversa, e o processo pode ser contínuo ou passo a passo.

I. INTRODUÇÃO

O presente trabalho descreve um ambiente interativo de programação com facilidades para criar, executar e depurar programas.

O trabalho objetiva estimular a concepção de programas com um alto nível de abstração, promovendo programação por refinamento passo a passo, poupando o usuário de frustrantes detalhes de sintaxe durante a escrita de programas.

-
- * Tecnólogo em Processamento de Dados (FURB-SC 1981)
 Professor (FURB - SC)
 Área de interesse: Engenharia de Software
- ** Bacharel em Ciências econômicas (UFSC - 1981)
 Área de interesse: Banco de Dados
- *** Tecnólogo em Processamento de Dados (UFRGS)
 Área de interesse: Inteligência Artificial
- **** Mestre e Doutor em Informática (PUC - RIO)
 Pesquisador e professor CPGCC-UFRGS.
 Área de interesse : Linguagens de Programação
- + Mestrandos em Ciência da Computação (UFRGS)

Os programas são criados na forma top-down pela inserção de novas declarações e expressões na posição do cursor com o esqueleto previamente moldado. Os moldes são gerados por comandos, mas expressões e declarações são construídas caracter a caracter, sendo impossível cometer erros em gabaritos, pois são pré-definidos.

Erros em textos de usuários são imediatamente detectados, pois são analisados pelo editor, frase por frase, dando ao usuário a certeza de que, ao finalizar a edição de um programa, o mesmo estará livre de erros de sintaxe.

Além da edição, o trabalho enfatiza a execução controlada de programas, que tanto pode ser na forma normal como na forma reversa. A execução poderá ser contínua ou passo a passo, permitindo assim que o usuário acompanhe a mudança de estados do programa.

A execução passo a passo permite ao usuário interromper o programa no ponto desejado. Na execução contínua esta parada será aleatória, caso não tenha sido pré-determinada.

Na execução reversa as ações são desfeitas passo a passo e as variáveis assumirão os valores possuídos em estados anteriores. Com isso, o usuário poderá retroceder a execução até onde desejar, corrigir erros nos comandos contidos na região onde as ações foram desfeitas, retomando, após a execução dos mesmos. Cabe neste ponto salientar o fato de que existem algumas ações que não podem ser desfeitas, como por exemplo aquelas que envolvem os comandos de impressão, comandos de transmissão de dados, etc.

II. INTERFACE COM O USUARIO

2.1- Execução normal

A execução de um programa é normalmente feita do início ao fim do mesmo, a menos que haja interferência do usuário no sentido de mudar este comportamento.

As opções de mudança que o usuário tem são:

- marcar um trecho para ser executado;
- definir o modo de execução e
- selecionar variáveis para serem acompanhadas.

No primeiro caso, a execução será do início do programa até o ponto marcado, quando será mostrado em destaque a instrução ou linha do programa, que está sendo executada. Caso exista alguma chamada de procedure dentro do trecho marcado, ela será totalmente executada mesmo estando fora do escopo definido.

Se o usuário, ao definir o modo de execução, optar por

"passo a passo", terá que teclar <ENTER> após a execução de cada instrução. No entanto, se a opção foi "execução contínua", poderá ainda determinar um tempo de espera (delay) entre uma instrução e outra.

Caso o usuário tenha selecionado algumas variáveis para serem acompanhadas, as mesmas serão mostradas em uma janela no vídeo, onde aparecerão todos os valores assumidos durante a execução do programa.

Durante o processo de execução, o usuário poderá interrompê-lo para fazer uma análise mais detalhada das variáveis naquele momento, e se desejar poderá alterar os valores das mesmas e retomar a execução.

2.2- Execução reversa

A execução reversa comporta-se de forma análoga à descrita na execução normal, quanto à seleção, modo de execução, seleção de variáveis e trocas de valores.

A opção por esta modalidade poderá ser feita a qualquer momento pelo usuário. Quando isto ocorrer, o sistema solicitará do usuário que informe o modo de execução, ou seja, se será contínua ou passo a passo. Caso a opção tenha sido "execução contínua", o sistema entrará em modo de edição e o usuário marcará o fim do trecho. Após isto, o sistema retornará a execução e o comportamento é o mesmo descrito anteriormente.

Se a opção for "passo a passo", o sistema automaticamente começará a execução e destacará a instrução que está sendo executada e esperará que o usuário tecle <ENTER> para continuar.

III. EXECUÇÃO CONTROLADA

3.1. Justificativa

O objetivo do controle da execução de um programa é permitir paradas, troca de valores de variáveis e execução reversa. Dentro do controle será criado um arquivo histórico onde serão guardados os valores das variáveis que tiverem seu conteúdo mudado durante a execução. Este arquivo será consultado na execução reversa, quando as ações serão desfeitas e às variáveis serão atribuídos os valores referentes a estados anteriores.

3.2. Descrição

No programa em execução serão atribuídos endereços a cada linha. Estes endereços farão parte do arquivo histórico.

No arquivo histórico serão guardadas basicamente três tipos de informações, que são: o endereço da linha atual, o valor da variável no estado anterior e o endereço da variável na tabela de variáveis.

Além do arquivo histórico, existe uma tabela de variáveis que contém o nome da variável e o seu valor corrente.

No decorrer da execução do programa, o arquivo histórico vai recebendo os valores alterados na tabela de variáveis, associados a informações que situam a atribuição no contexto, como o número da linha atual e endereço na tabela de variáveis.

O processo de execução reversa utiliza o arquivo histórico e a tabela de variáveis da seguinte forma:

- O usuário determina a extensão de um trecho do programa a retornar;
- De cada registro do arquivo histórico, obtém-se o endereço na tabela de variáveis e o conteúdo com o qual a tabela será atualizada;
- O processo é repetido até o final do trecho de programa que foi marcado pelo usuário.

No início do bloco é colocado um registro no arquivo histórico com o campo "número da linha do programa" com o endereço de início do bloco e os demais campos zerados. A utilidade deste artifício será comentado mais adiante em chamadas de blocos.

IV- MANUSEIO DE VARIÁVEIS NO ARQUIVO HISTÓRICO E TABELA DE VARIÁVEIS DURANTE A INTERPRETAÇÃO DO PROGRAMA

4.1- Chamadas de blocos ("PROCEDURES")

No início da execução de um bloco é criado no arquivo histórico um registro com o número da linha inicial do bloco ("PROCEDURE"), com os demais campos zerados (conteúdo anterior e endereço na tabela de variáveis). As variáveis que a "PROCEDURE" possui são também criadas (inclusive as variáveis parâmetros, caso existam). As variáveis parâmetros são inicializadas com o valor passado pela chamada. As variáveis locais da "PROCEDURE" serão inicializadas com zero ou vazio, conforme o tipo.

No final da "PROCEDURE", todas as informações sobre a mesma serão excluídas da tabela de variáveis e do arquivo histó-

rico, preservando apenas as alterações feitas nas variáveis globais que eventualmente tenham sido utilizadas pela "PROCEDURE". Se um bloco for declarado com parâmetros de entrada/saída, no final da execução deste bloco serão atualizadas as variáveis do bloco que o chamou (passando o valor atual dos parâmetros).

PROGRAMA EXEMPLO

```

1 PROGRAM exemplo2;
2 VAR a,
3     b,
4     c : INTEGER;
5 PROCEDURE proc1 (d,e:INTEGER);
6 VAR f : INTEGER;
7 BEGIN
8     IF d > e
9     THEN f := e
10    ELSE IF e > d
11    THEN f := d
12    ELSE f := 1;
13    c := (d + e) / f;
14 END;
15 BEGIN
16    a := 10;
17    b := 2;
18    proc1 (a,b);
19    c := c + 1;
20 END.

```

Situação da tabela de variáveis e do arquivo histórico após as execuções das linhas 1 a 4 (definição de variáveis) e 13 a 14 (programa principal):

TABELA DE VARIÁVEIS

Sequência	Nome da variável	Conteúdo corrente
1	a	10
2	b	2
3	c	0

ARQUIVO HISTÓRICO

Número da linha do programa	Conteúdo anterior da variável	endereço na tab. de variáveis
1	0	0
1	0	1
1	0	2
1	0	3
13	0	1
14	0	2

Situação da tabela de variáveis e do arquivo histórico na continuação da execução da linha 15 (chamada de "PROCEDURE") e após as execuções das linhas 5 a 12:

TABELA DE VARIÁVEIS

Sequência	Nome da variável	Conteúdo corrente
1	a	10
2	b	2
3	c	6
4	d	10
5	e	2
6	f	2

ARQUIVO HISTÓRICO

Número da linha do programa	Conteúdo anterior da variável	endereço na tab. de variáveis
1	0	0
1	0	1
1	0	2
1	0	3
13	0	1
14	0	2
5	0	0 (início de bloco)
5	0	4
5	0	5
5	0	6
8	0	6
12	0	3 (variável global)

Situação das tabelas após a continuação da execução (a partir da linha 15) até o final (linha 16):

TABELA DE VARIÁVEIS

Sequência	Nome da variável	Conteúdo corrente
1	a	10
2	b	2
3	c	8

ARQUIVO HISTÓRICO

Número da linha do programa	Conteúdo anterior da variável	endereço na tab. de variáveis
1	0	0
1	0	1
1	0	2
1	0	3
13	0	1
14	0	2
15	0	3 (variável global)
16	7	3

V- MANUSEIO DE VARIÁVEIS NO ARQUIVO HISTÓRICO E TABELA DE VARIÁVEIS DURANTE A INTERPRETAÇÃO REVERSA

A execução reversa consiste em desfazer ações semânticas, o que é feito com o auxílio do arquivo histórico.

O retorno ao estado anterior (último comando processado) é feito simplesmente consultando o arquivo histórico, o qual contém informações que dispensam a análise do programa fonte.

A medida que as ações forem sendo desfeitas, as informações sobre elas serão colocadas na tabela de variáveis e excluídas do arquivo histórico.

Cada registro do arquivo histórico contém informações sobre uma variável que tem seu conteúdo mudado durante a execução de um comando, ou sobre variáveis declaradas no início do bloco.

Se forem informações sobre comandos que alteram variáveis, será feita uma atualização na tabela de variáveis, sendo que para cada operação é considerado um passo de retorno.

Se forem informações sobre variáveis declaradas no início de bloco, a tabela de variáveis será atualizada tantas vezes quantas forem as variáveis declaradas, e será considerado um único passo de retorno. O reconhecimento deste fato é caracterizado pela presença contígua nos registros anteriores do mesmo endereço no campo "endereço da linha atual" no arquivo histórico.

Quando na execução reversa de um comando de laço ("while" do Pascal), se o bloco for marcado no comando que o inicia (início do "while") e for escolhido a opção "execução reversa continua", todos os laços serão desfeitos automaticamente. Caso haja interesse de parar no meio do laço, interrompe-se e escolhe-se a opção passo a passo para desfazer e ou executar, até encontrar a situação desejada.

VI- CONCLUSÃO

Este trabalho descreve um método para execução normal e reversa de um programa em Pascal, prevendo facilidades de controle sobre a mesma, facilitando a depuração de programas pelo usuário.

Até o momento foi implementado um protótipo usando um subconjunto da linguagem Pascal, onde se pode comprovar a validade do método.

As estruturas foram projetadas objetivando simplificar a execução normal ou reversa, alocando o mínimo necessário de memória, só sendo guardadas informações estritamente necessárias para o gerenciamento da depuração, conseguindo-se desta forma depurar programas maiores e com uma boa performance.

VII - BIBLIOGRAFIA

1. AHO, A.V.; ULLMAN J.D.- Principles of Compiler Design, Addison-Wesley Publishing Company, 1979.
2. SETZER V.W.; MELO I.S.H.- A Construção de um Compilador, Editora Campus Ltda, Rio de Janeiro, 1983.