**Chamada de procedimentos iterativos**

|  |  |  |
| --- | --- | --- |
| Algoritmo **Exemplo2** | INPP | L4:NADA |
| variaveis | AMEM 2 | CRVL 1 |
| inteiro x, y; | DSVS L2 | CRVL 2 |
| procedimento p(vazio) | L1:NADA | MULT |
| variaveis | AMEM 1 | ARMZ 1 |
| z inteiro; | CRVL 0 | **RTPR** |
| inicio | ARMZ 3 | L2:NADA |
| z := x; | CRVL 0 | CRCT 4 |
| x := x-1; | CRCT 1 | ARMAZ 0 |
| se z > 1 entao | SUBT | **CHPR L1** |
| p;//CHPR L1 | ARMZ 0 | CRVL 0#ER1 |
| senao | CRVL 3 | IMPR |
| y := 1; | CRCT 1 | CRVL 1 |
| fim se | CMMA | IMPR |
| y := y\*z;//**DSVS L5** | DSVF L3 |  |
| fim procedimento | **CHPR L1** |  |
| inicio | DSVS L4 |  |
| x := 4; | L3:NADA |  |
| p; //CHPR L1 | CRCT 1 |  |
| imprima (x,y); | ARMZ 1 |  |
| fim |  |  |

CHPR p (chama procedimento)

s <- s+1

M[s] <- i+1

i <- p

RTPR (retorna do procedimento)

i <- M[S]

S <- S-1

DMEM n (desaloca memória)

S <- S-n

**Traduzir o Exemplo2 .....**

**Chamada de Procedimentos Recursivos**

Suponha que programa **Exemplo2** seja reescrito com uma chamada recursiva dentro do procedimento **P**, como resolver ?

Algoritmo **Exemplo3**

variaveis

x, y;

procedimento p(vazio)

variaveis

z inteiro;

inicio

z := x;

x := x-1;

se z > 1 entao

p();

senao

y := 1;

fim se

y := y\*z;

fim procedimento

inicio

x := 4;

p;

imprima (x,y);

fim

Até o momento trabalhamos com endereço fixo para as variáveis, entretanto, como visto na pilha de execução do **Exemplo3** a variável **z** deve possuir endereços diferentes conforme a profundidade de recursão.

O endereço fixo será substituído pelo endereço relativo, o compilador associará a cada chamada de um procedimento, um valor para um **registrador base**, as variáveis locais de um procedimento terão um deslocamento fixo a partir desse registrador.

**Registrador Base:**

D[0] – O programa principal é um procedimento de nível 0 e suas variáveis globais estão armazenadas a partir do endereço em D[0].

D[1] – Todo procedimento ou função é de nível 1, e suas variáveis locais e parâmetros são acessados a partir do endereço em D[1]

**Modificação das instruções para tratar variáveis globais e locais**

INPP (Iniciar Procedimento Principal)

s <- -1

D[0] <- 0

D[1] <- 0

ENPR (Entrada de procedimento)

s <- s+1

M[s] <- D[1]

D[1] <- s+1

Essa instrução será a primeira instrução de um procedimento, note que ela coloca em D[1] o endereço onde deverá estar a primeira variável local do procedimento, caso tenha variável local, este espaço será reservado pela instrução AMEM que viria a seguir.

RTPR (Retornar procedimento)

D[1] <- M[s] // Endereço anterior em D[1]

i <- M[s-1] // Linha abaixo da chamada de proc.

s <- s-2

A instrução RTPR restaura o conteúdo do registrador base que foi empilhado pela a instrução ENPR, a instrução é chamada após DMEM

CRVL m,n

s <- s+1

M[s] <- M[D[m] + n]

ARMZ m,n

M[D[m]+n] <- M[s]

s <- s-1

**traduzir o Exemplo3 com procedimento......**

**Procedimento com parâmetro por valor**

Consideramos agora que um procedimento tenha parâmetros passados por valor.

P( E1, E2, ...,EN)

Se o procedimento tem N parâmetros, então o i-ésimo parâmetro na memória **M** e terá deslocamento – (N+3-i)

Exemplo

P(a,b,c)

O endereço dos parâmetros terão valores NEGATIVO:

c=1,-3

b=1,-4

a=1,-5

Agora teremos que modificar a instrução RTPR de tal forma que ela remova os parâmetros da pilha ao sair do procedimento, onde n é o número de parâmetros

RTPR n

D[1] <- M[s]

i <- M[s-1]

S <- S-(n+2)

Algoritmo **Exemplo3b**

variaveis

x inteiro;

procedimento P(inteiro a)

variaveis

z inteiro;

inicio

z := a;

se z > 1 entao

p(a-1);

senao

x := 1;

fim-se;

x := x\*z;

fim

inicio

p(4);

imprima(x);

fim

**traduzir o Exemplo3 com procedimento com parâmetros ......**

**Exercício:**

**-------------**

a) O que o programa em MEPA faz? Qual é a sua saída?

b) Escreva um programa em na sintaxe do trabalho 2 o programa em MEPA.

INPP

AMEM 1

DSVS L1

L2:NADA

ENPR

AMEM 1

CRVL 1,-3

CRCT 1

CMIG

DSVF L3

CRVL 1,-3

ARMZ 0,0

DSVS L4

L3:NADA

CRVL 1,-3

CRCT 1

SUBT

CHPR L2

CRVL 1,-3

CRVL 0,0

MULT

ARMZ 0,0

L4:NADA

DMEM 1

RTPR 1

L1:NADA

CRCT 4

CHPR L2

CRVL 0,0

IMPR

DMEM 1

PARA