Exec.MEPA

Exec.Pascal Projeto

Construção de Compiladores Período Especial Aula 8: Comando While

Bruno Müller Junior

Departamento de Informática **UFPR**

2020



- Sintaxe
- Pluxo

- O Desvio na MEPA
- TraduçãoModelo
 - Esquema de Tradução
 - Exemplo
 - Exemplo
- **5** Exec.MEPA
 - Exec.MEPA
 - Primeira Iteração
 - Segunda Iteração
 - Terceira Iteração
- 6 Exec.Pascal
 - Execução
 - Execução



Fluxo

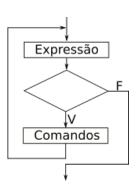
- O comando while (regra 23) é uma das possibilidades de <comando sem rótulo> (regra 18).
- O analisador sintático irá escolher a regra 18 quando estiver na regra 23 e encontrar o token while.

```
18. <comando sem rótulo> ::= <atribuição> | ... | <comando repetitivo> 23. <comando repetitivo> ::= WHILE <expressão> D0 <comando sem rótulo>
```



Fluxo

- O fluxo de execução do comando while não é sequencial.
- Existem desvios do fluxo para locais específicos.
- Os desvios são de dois tipos: incondicionais e condicionais.



Desvio na MEPA

 Para implementar a mudança de fluxo, a MEPA contém duas instruções:

Instrução	Ação	Significado
DSVS p	i:=p;	Desvia
		Sempre
DSVF p	Se (M[s]==0) então i:=p;	Desvia se
	senão i:=i+1	Falso
	s:=s-1	

- Em ambas o parâmetro é o endereço de desvio do fluxo, chamado rótulo.
- Por convenção deste material, os rótulos seguem sempre o padrão da letra Rdd, onde dd são dois dígitos. Exemplos: R00, R01,

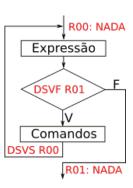


Exec.MEPA

Sintaxe

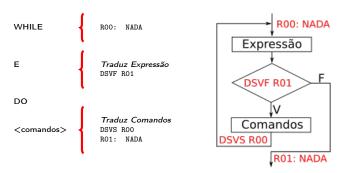
Modelo

- A tradução do programa para MEPA é simples:
 - inclui-se rótulos nas entradas de fluxo:
 - inclui-se desvios condicionais analisando o resultado da expressão.
 - inclui-se desvios incondicionais ao final do comando.



Esquema de Tradução

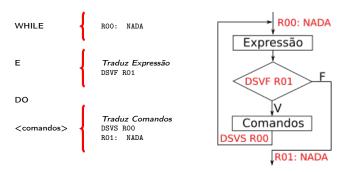
Outra forma de ver é através do esquema de tradução;



Esquema de Tradução

• Que já sugere a geração de código:

23. <comando repetitivo> ::= WHILE <expressão> DO <comando sem rótulo>



```
program comandoWhile;
var n, k: integer;
    f1, f2, f3:integer;
begin
    read(n)
    f1:=0; f2:=1; k:=1;
    while (k <= n) do
    begin
    f3:=f2+f1;
    f1:=f2;
    f2:=f3;
    k:=k+1
    end;
    write(n,k)
    end.</pre>
```

```
program comandoWhile;
var n, k: integer;
    f1, f2, f3:integer;
begin
    read(n)
    f1:=0; f2:=1; k:=1;
    while (k <= n) do
    begin
        f3:=f2+f1;
        f1:=f2;
        f2:=f3;
        k:=k+1
    end;
    write(n,k)
    end.</pre>
```

INPP

```
program comandoWhile;
var n, k: integer;
f1, f2, f3:integer;
begin
read(n)
f1:=0; f2:=1; k:=1;
while (k <= n) do
begin
f3:=f2+f1;
f1:=f2;
f2:=f3;
k:=k+1
end;
write(n,k)
end.</pre>
```

INPP



```
program comandoWhile;
                                       INPP
var n, k: integer;
    f1, f2, f3:integer;
begin
   read(n)
   f1:=0: f2:=1: k:=1:
   while (k <= n) do
   begin
      f3:=f2+f1:
      f1:=f2:
      f2:=f3;
      k:=k+1
   end:
   write(n,k)
   end.
```

```
VS
              [0,0,?]
  n
Símb.
        Cat. Infos
```

INPP

```
program comandoWhile;
var n, k: integer;
f1, f2, f3:integer;
begin
    read(n)
f1:=0; f2:=1; k:=1;
while (k <= n) do
    begin
    f3:=f2+f1;
    f1:=f2;
    f2:=f3;
    k:=k+1
end;
write(n,k)
end.</pre>
```

```
k
          VS
               [0.1.?]
          VS
                [0,0,?]
  n
Símb.
         Cat.
               Infos
```

```
program comandoWhile;
                                       INPP
var n, k: integer;
                                       AMEM 2
    f1, f2, f3:integer;
begin
   read(n)
   f1:=0: f2:=1: k:=1:
   while (k <= n) do
   begin
      f3:=f2+f1:
      f1:=f2:
      f2:=f3;
      k:=k+1
   end:
   write(n,k)
   end.
```

```
k
         VS
               [0.1.int]
         VS
               [0,0,int]
  n
Símb.
        Cat.
              Infos
```

```
program comandoWhile;
                                       INPP
var n, k: integer;
                                       AMEM 2
    f1, f2, f3:integer;
                                       AMEM 3
begin
   read(n)
   f1:=0: f2:=1: k:=1:
   while (k <= n) do
   begin
      f3:=f2+f1:
      f1:=f2:
      f2:=f3;
      k:=k+1
   end:
   write(n,k)
   end.
```

```
[0,4,int]
  f3
          VS
  f2
                [0,3,int]
  f1
                [0.2.int]
          VS
  k
          VS
               [0.1.int]
          VS
                [0,0,int]
  n
Símb.
        Cat.
               Infos
```

```
program comandoWhile;
                                       INPP
var n, k: integer;
                                       AMEM 2
    f1, f2, f3:integer;
                                       AMEM 3
begin
   read(n)
   f1:=0: f2:=1: k:=1:
   while (k <= n) do
   begin
      f3:=f2+f1:
      f1:=f2:
      f2:=f3;
      k:=k+1
   end:
   write(n,k)
   end.
```

```
[0.4.int]
  f3
          VS
  f2
                [0,3,int]
  f1
                [0.2.int]
          VS
  k
          VS
               [0.1.int]
          VS
                [0,0,int]
  n
Símb.
        Cat.
               Infos
```

```
program comandoWhile;
var n, k: integer;
f1, f2, f3:integer;
begin
    read(n)
f1:=0; f2:=1; k:=1;
while (k <= n) do
begin
    f3:=f2+f1;
    f1:=f2;
    f2:=f3;
    k:=k+1
end;
write(n,k)
end.</pre>
```

```
INPP
AMEM 2
AMEM 3
LEIT
ARMZ 0,0
```

```
[0.4.int]
  f3
          VS
  f2
                [0,3,int]
  f1
                [0.2.int]
          VS
  k
          VS
               [0.1.int]
          VS
                [0,0,int]
  n
Símb.
        Cat.
               Infos
```

```
program comandoWhile;
                                       INPP
var n, k: integer;
                                       AMEM 2
    f1, f2, f3:integer;
                                       AMEM 3
begin
                                       LEIT
   read(n)
                                       ARMZ 0,0
   f1:=0: f2:=1: k:=1:
                                       CRCT 0
   while (k <= n) do
                                       ARMZ 0,2
   begin
      f3:=f2+f1:
      f1:=f2:
      f2:=f3;
      k:=k+1
   end:
   write(n,k)
   end.
```

```
[0.4.int]
  f3
          VS
  f2
                [0,3,int]
                [0.2.int]
  f1
          VS
  k
          VS
               [0.1.int]
          VS
                [0,0,int]
  n
Símb.
        Cat.
               Infos
```

```
program comandoWhile;
                                       INPP
var n, k: integer;
                                       AMEM 2
    f1, f2, f3:integer;
                                       AMEM 3
begin
                                       LEIT
   read(n)
                                       ARMZ 0,0
   f1:=0: f2:=1: k:=1:
                                       CRCT 0
   while (k <= n) do
                                       ARMZ 0,2
                                       CRCT 1
   begin
      f3:=f2+f1:
                                       ARMZ 0.3
      f1:=f2:
      f2:=f3;
      k:=k+1
   end:
   write(n.k)
   end.
```

```
[0.4.int]
  f3
          VS
  f2
                [0,3,int]
                [0.2.int]
  f1
          VS
  k
          VS
               [0.1.int]
          VS
                [0,0,int]
  n
Símb.
        Cat.
               Infos
```

```
program comandoWhile;
var n, k: integer;
    f1, f2, f3:integer;
begin
    read(n)
    f1:=0; f2:=1; k:=1;
    while (k <= n) do
    begin
    f3:=f2+f1;
    f1:=f2;
    f2:=f3;
    k:=k+1
    end;
    write(n,k)
    end.</pre>
```

```
INPP
AMEM 2
AMEM 3
LEIT
ARMZ 0,0
CRCT 0
ARMZ 0,2
CRCT 1
ARMZ 0,3
CRCT 1
ARMZ 0,1
```

```
[0.4.int]
  f3
          VS
  f2
               [0,3,int]
               [0.2.int]
  f1
          VS
  k
          VS
               [0.1.int]
          VS
               [0,0,int]
  n
Símb.
        Cat.
               Infos
```

Exec.MEPA
00000000000
00000000
00000000

Exemplo

```
program comandoWhile;
                                       INPP
var n, k: integer;
                                       AMEM 2
    f1, f2, f3:integer;
                                       AMEM 3
begin
                                       LEIT
   read(n)
                                       ARMZ 0,0
   f1:=0: f2:=1: k:=1:
                                       CRCT 0
   while (k <= n) do
                                       ARMZ 0,2
                                       CRCT 1
   begin
      f3:=f2+f1:
                                       ARMZ 0.3
      f1:=f2:
                                       CRCT 1
      f2:=f3;
                                       ARMZ 0,1
      k:=k+1
                                   ROO: NADA
   end:
   write(n.k)
   end.
```

```
[0.4.int]
  f3
          VS
  f2
               [0,3,int]
               [0.2.int]
  f1
          VS
  k
          VS
               [0.1.int]
          VS
               [0,0,int]
  n
Símb.
        Cat.
               Infos
```

xec.MEPA 000000000 000000 000000 000000

Exemplo

```
program comandoWhile;
                                       INPP
var n, k: integer;
                                       AMEM 2
    f1, f2, f3:integer;
                                       AMEM 3
begin
                                       LEIT
   read(n)
                                       ARMZ 0,0
   f1:=0: f2:=1: k:=1:
                                       CRCT 0
   while (k \le n) do
                                       ARMZ 0,2
                                       CRCT 1
   begin
      f3:=f2+f1:
                                       ARMZ 0.3
      f1:=f2:
                                       CRCT 1
      f2:=f3;
                                       ARMZ 0,1
      k:=k+1
                                   ROO:NADA
   end:
                                       CRVL 0.1
   write(n.k)
                                       CRVL 0,0
   end.
                                       CMEG
```

```
[0.4.int]
  f3
          VS
  f2
               [0,3,int]
               [0.2.int]
  f1
          VS
  k
          VS
               [0.1.int]
          VS
               [0,0,int]
  n
Símb.
        Cat.
               Infos
```

xec.MEPA 0000000000 000000 0000000 0000000

Exemplo

```
program comandoWhile;
                                       INPP
var n, k: integer;
                                       AMEM 2
    f1, f2, f3:integer;
                                       AMEM 3
begin
                                       LEIT
   read(n)
                                       ARMZ 0,0
   f1:=0: f2:=1: k:=1:
                                       CRCT 0
   while (k \le n) do
                                       ARMZ 0,2
                                       CRCT 1
   begin
      f3:=f2+f1:
                                       ARMZ 0.3
      f1:=f2:
                                       CRCT 1
      f2:=f3;
                                       ARMZ 0,1
      k:=k+1
                                   ROO:NADA
   end:
                                       CRVL 0.1
   write(n.k)
                                       CRVL 0,0
   end.
                                       CMEG
                                       DSVF RO1
```

```
[0.4.int]
  f3
          VS
  f2
               [0,3,int]
               [0.2.int]
  f1
          VS
  k
          VS
               [0.1.int]
          VS
               [0,0,int]
  n
Símb.
        Cat.
               Infos
```

xec.MEPA 0000000000 0000000 0000000

Exemplo

```
program comandoWhile;
                                       INPP
var n, k: integer;
                                       AMEM 2
    f1, f2, f3:integer;
                                       AMEM 3
begin
                                       LEIT
   read(n)
                                       ARMZ 0,0
   f1:=0: f2:=1: k:=1:
                                       CRCT 0
   while (k <= n) do
                                       ARMZ 0,2
                                       CRCT 1
   begin
      f3:=f2+f1:
                                       ARMZ 0.3
      f1:=f2:
                                       CRCT 1
      f2:=f3;
                                       ARMZ 0,1
      k:=k+1
                                   ROO:NADA
   end:
                                       CRVL 0.1
   write(n.k)
                                       CRVL 0,0
   end.
                                       CMEG
                                       DSVF RO1
                                       CRVL 0,3
                                       CRVL 0,2
                                       SOMA
                                       ARMZ 0.4
```

```
[0.4.int]
  f3
          VS
  f2
                [0,3,int]
  f1
          VS
                [0.2.int]
  k
          VS
                [0.1.int]
          VS
                [0,0,int]
  n
Símb.
        Cat.
               Infos
```

```
program comandoWhile;
                                       INPP
                                                       CRVL 0,3
var n, k: integer;
                                       AMEM 2
                                                       ARMZ 0.2
    f1, f2, f3:integer:
                                       AMEM 3
begin
                                       LEIT
                                                       CRVL 0,4
   read(n)
                                       ARMZ 0,0
                                                       ARMZ 0,3
   f1:=0: f2:=1: k:=1:
                                       CRCT 0
   while (k <= n) do
                                       ARMZ 0,2
                                                       CRVL 0,1
                                       CRCT 1
                                                       CRCT 1
   begin
      f3:=f2+f1:
                                       ARMZ 0.3
                                                       ARMZ 0.1
      f1:=f2:
                                       CRCT 1
      f2:=f3;
                                       ARMZ 0,1
      k:=k+1
                                   ROO:NADA
   end:
                                       CRVL 0.1
   write(n.k)
                                       CRVL 0,0
   end.
                                       CMEG
                                       DSVF RO1
                                       CRVL 0,3
                                       CRVL 0,2
                                       SOMA
                                       ARMZ 0.4
```

```
f3
          VS
                [0.4.int]
  f2
                [0,3,int]
  f1
          VS
                [0.2.int]
  k
          VS
                [0.1.int]
          VS
                [0,0,int]
  n
Símb.
        Cat.
               Infos
```

```
INPP
                                                       CRVL 0,3
program comandoWhile;
var n. k: integer:
                                       AMEM 2
                                                       ARMZ 0.2
    f1, f2, f3:integer:
                                       AMEM 3
begin
                                       LEIT
                                                       CRVL 0,4
   read(n)
                                       ARMZ 0,0
                                                       ARMZ 0,3
   f1:=0: f2:=1: k:=1:
                                       CRCT 0
   while (k <= n) do
                                       ARMZ 0,2
                                                       CRVL 0,1
                                       CRCT 1
                                                       CRCT 1
   begin
      f3:=f2+f1:
                                       ARMZ 0.3
                                                       ARMZ 0.1
      f1:=f2:
                                       CRCT 1
                                                       DSVS ROO
      f2:=f3;
                                       ARMZ 0,1
                                                 RO1: NADA
      k:=k+1
                                  ROO:NADA
   end:
                                       CRVL 0.1
   write(n.k)
                                       CRVL 0,0
   end.
                                       CMEG
                                       DSVF RO1
                                       CRVL 0,3
                                       CRVL 0,2
                                       SOMA
                                       ARMZ 0.4
```

```
f3
          VS
                [0.4.int]
  f2
                [0,3,int]
  f1
          VS
                [0.2.int]
  k
          VS
                [0.1.int]
          VS
                [0,0,int]
  n
Símb.
        Cat.
               Infos
```

Exec.MEPA
0000000000
00000000
00000000

Exemplo

Sintaxe

```
INPP
                                                       CRVL 0,3
program comandoWhile;
var n. k: integer:
                                       AMEM 2
                                                       ARMZ 0.2
    f1, f2, f3:integer:
                                       AMEM 3
begin
                                       LEIT
                                                       CRVL 0,4
   read(n)
                                       ARMZ 0,0
                                                       ARMZ 0,3
   f1:=0: f2:=1: k:=1:
                                       CRCT 0
   while (k <= n) do
                                       ARMZ 0,2
                                                       CRVL 0,1
                                       CRCT 1
                                                       CRCT 1
   begin
      f3:=f2+f1:
                                       ARMZ 0.3
                                                       ARMZ 0.1
      f1:=f2:
                                       CRCT 1
                                                       DSVS ROO
      f2:=f3;
                                       ARMZ 0,1
                                                 RO1: NADA
      k:=k+1
                                   ROO:NADA
                                                       CRVL 0,0
   end:
                                       CRVL 0.1
                                                       IMPR
   write(n.k)
                                                       CRVL 0,1
                                       CRVL 0,0
   end.
                                       CMEG
                                                       TMPR.
                                       DSVF RO1
                                       CRVL 0,3
                                       CRVL 0,2
                                       SOMA
```

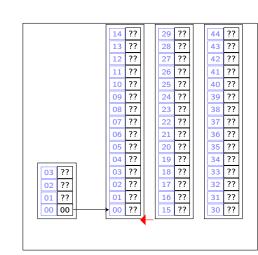
ARMZ 0.4

```
f3
          VS
                [0.4.int]
  f2
                [0,3,int]
  f1
          VS
                [0.2.int]
  k
          VS
                [0.1.int]
          VS
                [0,0,int]
  n
Símb.
        Cat.
               Infos
```

```
INPP
program comandoWhile;
                                                       CRVL 0,3
var n, k: integer;
                                       AMEM 2
                                                       ARMZ 0.2
    f1, f2, f3:integer:
                                       AMEM 3
begin
                                       LEIT
                                                       CRVL 0,4
   read(n)
                                       ARMZ 0,0
                                                       ARMZ 0,3
   f1:=0: f2:=1: k:=1:
                                       CRCT 0
   while (k <= n) do
                                       ARMZ 0,2
                                                       CRVL 0,1
                                       CRCT 1
                                                       CRCT 1
   begin
      f3:=f2+f1:
                                       ARMZ 0.3
                                                       ARMZ 0.1
      f1:=f2:
                                       CRCT 1
                                                       DSVS ROO
      f2:=f3;
                                       ARMZ 0,1
                                                 RO1: NADA
      k:=k+1
                                   ROO: NADA
                                                       CRVL 0.0
   end:
                                       CRVL 0.1
                                                       IMPR
   write(n.k)
                                       CRVL 0,0
                                                       CRVL 0,1
   end.
                                       CMEG
                                                       TMPR.
                                                       DMEM 5
                                       DSVF RO1
                                       CRVL 0,3
                                                       PARA
                                       CRVL 0,2
                                       SOMA
                                       ARMZ 0.4
```

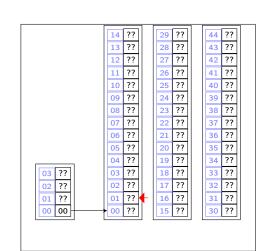
Símb. Cat. Infos

INPP			CRVL	0,3
AMEM	2		ARMZ	0,2
AMEM	3			
LEIT			CRVL	0,4
ARMZ	0,0		ARMZ	0,3
CRCT	0			
ARMZ	0,2		CRVL	0,1
CRCT	1		CRCT	1
ARMZ	0,3		ARMZ	0,1
CRCT	1		DSVS	R00
ARMZ	0,1	R01:	NADA	
ROO:NADA			CRVL	0,0
CRVL	0,1		IMPR	
CRVL	0,0		CRVL	0,1
CMEG			IMPR	
DSVF	R01		DMEM	5
CRVL	0,3		PARA	
CRVL	0,2			
SOMA				



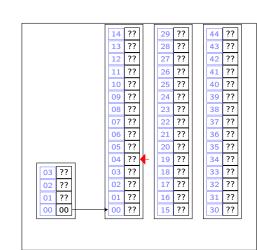
Traducão

```
INPP
                    CRVL 0,3
    AMEM 2
                    ARMZ 0.2
    AMEM 3
    LEIT
                    CRVL 0,4
    ARMZ 0.0
                    ARMZ 0.3
    CRCT 0
    ARMZ 0,2
                    CRVL 0,1
    CRCT 1
                    CRCT 1
    ARMZ 0.3
                    ARMZ 0.1
    CRCT 1
                    DSVS ROO
    ARMZ 0,1
              RO1: NADA
ROO: NADA
                    CRVL 0.0
    CRVL 0,1
                    IMPR
    CRVL 0,0
                    CRVL 0,1
    CMEG
                    IMPR
                    DMEM 5
    DSVF R01
    CRVL 0,3
                    PARA
    CRVL 0.2
    SOMA
```

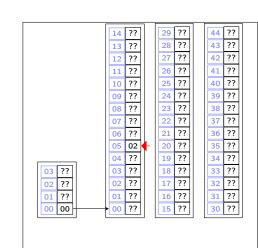


ARMZ 0,4

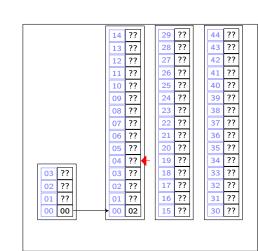
INPP			CRVL	0,3
AMEM	2		ARMZ	0,2
AMEM	3			
LEIT			CRVL	0,4
ARMZ	0,0		ARMZ	0,3
CRCT	0			
ARMZ	0,2		CRVL	0,1
CRCT	1		CRCT	1
ARMZ	0,3		ARMZ	0,1
CRCT	1		DSVS	R00
ARMZ	0,1	R01:	NADA	
ROO:NADA			CRVL	0,0
CRVL	0,1		IMPR	
CRVL	0,0		CRVL	0,1
CMEG			IMPR	
DSVF	R01		DMEM	5
CRVL	0,3		PARA	
CRVL	0,2			
SOMA				
ARMZ	0,4			



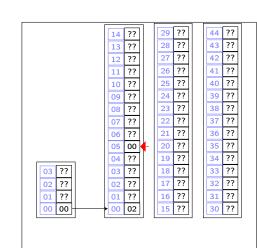
INPP			CRVL	0,3
AMEM	2		ARMZ	0,2
AMEM	3			
LEIT			CRVL	0,4
ARMZ	0,0		ARMZ	0,3
CRCT	0			
ARMZ	0,2		CRVL	0,1
CRCT	1		CRCT	1
ARMZ	0,3		ARMZ	0,1
CRCT	1		DSVS	R00
ARMZ	0,1	R01:	NADA	
ROO:NADA			CRVL	0,0
CRVL	0,1		IMPR	
CRVL	0,0		CRVL	0,1
CMEG			IMPR	
DSVF	R01		DMEM	5
CRVL	0,3		PARA	
CRVL	0,2			
SOMA				
ARMZ	0.4			



		CRVL	0,3
2		ARMZ	0,2
3			
		CRVL	0,4
0,0		ARMZ	0,3
0			
0,2		CRVL	0,1
1		CRCT	1
0,3		ARMZ	0,1
1		DSVS	R00
0,1	R01:	NADA	
		CRVL	0,0
0,1		IMPR	
0,0		CRVL	0,1
		IMPR	
R01		DMEM	5
0,3		PARA	
0,2			
0,2			
	1 0,3 1 0,1 0,1	0,0 0 0,2 1 0,3 1 0,1 R01:	2 ARMZ 3 CRVL 0,0 ARMZ 0 CRVL 1 CRCT 0,3 ARMZ 1 DSVS 0,1 RO1: NADA CRVL 0,1 IMPR 0,0 CRVL IMPR RO1 DMEM

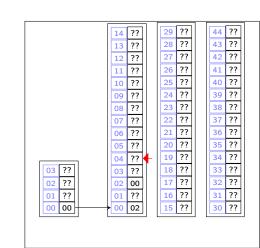


	INPP			CRVL	0,3
	AMEM	2		ARMZ	0,2
	AMEM	3			
	LEIT			CRVL	0,4
	ARMZ	0,0		ARMZ	0,3
	CRCT	0			
	ARMZ	0,2		CRVL	0,1
	CRCT	1		CRCT	1
	ARMZ	0,3		ARMZ	0,1
	CRCT	1		DSVS	R00
	ARMZ	0,1	R01:	NADA	
R00:	NADA			CRVL	0,0
	CRVL	0,1		IMPR	
	CRVL	0,0		CRVL	0,1
	CMEG			IMPR	
	DSVF	R01		DMEM	5
	CRVL	0,3		PARA	
	CRVL	0,2			
	SOMA				
	ARMZ	0,4			



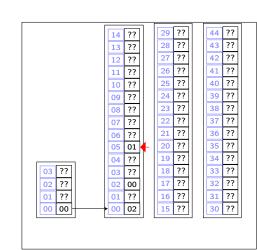
Tradução 0 00 0000000000 0000000000

```
INPP
                    CRVL 0,3
    AMEM 2
                    ARMZ 0.2
    AMEM 3
    LEIT
                    CRVL 0,4
    ARMZ 0.0
                    ARMZ 0.3
    CRCT 0
    ARMZ 0,2
                    CRVL 0,1
    CRCT 1
                    CRCT 1
    ARMZ 0.3
                    ARMZ 0.1
    CRCT 1
                    DSVS ROO
    ARMZ 0,1
              RO1: NADA
ROO: NADA
                    CRVL 0.0
    CRVL 0,1
                    IMPR
    CRVL 0,0
                    CRVL 0,1
    CMEG
                    IMPR
                    DMEM 5
    DSVF R01
    CRVL 0,3
                    PARA
    CRVL 0.2
    SOMA
```

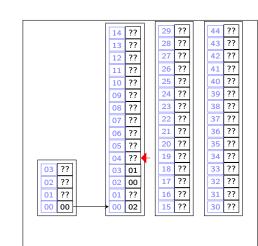


ARMZ 0,4

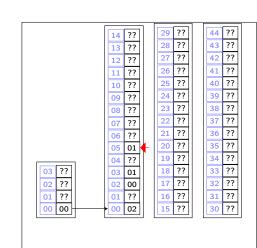
	INPP			CRVL	0,3
	AMEM	2		ARMZ	0,2
	AMEM	3			
	LEIT			CRVL	0,4
	ARMZ	0,0		ARMZ	0,3
	CRCT	0			
	ARMZ	0,2		CRVL	0,1
	CRCT	1		CRCT	1
	ARMZ	0,3		ARMZ	0,1
	CRCT	1		DSVS	R00
	ARMZ	0,1	R01:	NADA	
R00:	NADA			CRVL	0,0
	CRVL	0,1		IMPR	
	CRVL	0,0		CRVL	0,1
	CMEG			IMPR	
	DSVF	R01		DMEM	5
	CRVL	0,3		PARA	
	CRVL	0,2			
	SOMA				
	ARMZ	0,4			



INPP			CRVL	0,3
AMEM	2		ARMZ	0,2
AMEM	3			
LEIT			CRVL	0,4
ARMZ	0,0		ARMZ	0,3
CRCT	0			
ARMZ	0,2		CRVL	0,1
CRCT	1		CRCT	1
ARMZ	0,3		ARMZ	0,1
CRCT	1		DSVS	R00
ARMZ	0,1	RO1:	NADA	
ROO:NADA			CRVL	0,0
CRVL	0,1		IMPR	
CRVL	0,0		CRVL	0,1
CMEG			IMPR	
DSVF	R01		DMEM	5
CRVL	0,3		PARA	
CRVL	0,2			
SOMA				
ARMZ	0,4			



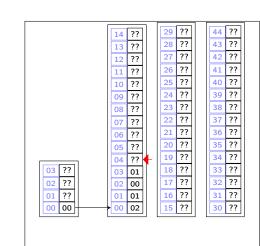
NPP			CRVL	0,3
MEM	2		ARMZ	0,2
MEM	3			
EIT			CRVL	0,4
RMZ	0,0		ARMZ	0,3
RCT	0			
RMZ	0,2		CRVL	0,1
RCT	1		CRCT	1
RMZ	0,3		ARMZ	0,1
RCT	1		DSVS	R00
RMZ	0,1	R01:	NADA	
ADA			CRVL	0,0
RVL	0,1		IMPR	
RVL	0,0		CRVL	0,1
MEG			IMPR	
SVF	R01		DMEM	5
RVL	0,3		PARA	
RVL	0,2			
AMO				
RMZ	0,4			
	MEM MEM EIT RMZ RCT RMZ RCT RMZ RCT RMZ RCT RMZ ADA RVL RVL MEG SVF RVL RVL OMA	MEM 2 MEM 3 EIT RMZ 0,0 RCT 0 RMZ 0,2 RCT 1 RMZ 0,1 ADA RVL 0,1 RVL 0,1 RVL 0,0 MEG RVL 0,3 RVL 0,2	MEM 2 MEM 3 EIT RMZ 0,0 RCT 0 RMZ 0,2 RCT 1 RMZ 0,3 RCT 1 RMZ 0,3 RCT 1 RMZ 0,1 RO1: ADA RVL 0,1 RVL 0,0 MEG SVF R01 RVL 0,3 RVL 0,2 OMA	MEM 2 ARMZ MEM 3 EIT CRVL RMZ 0,0 ARMZ RCT 0 RMZ RCT 1 CRVL RCT 1 DFVS RVZ 0,1 RO1: NADA ADA CRVL RVL 0,1 IMPR RVL 0,0 CRVL MEG IMPR SVF R01 DMEM RVL 0,3 PARA RVL 0,2 OMA



Exec.Pascal

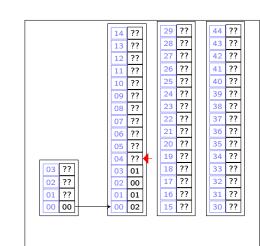
Exec.MEPA

	INPP			CRVL	0,3
	AMEM	2		ARMZ	0,2
	AMEM	3			
	LEIT			CRVL	0,4
	ARMZ	0,0		ARMZ	0,3
	CRCT	0			
	ARMZ	0,2		CRVL	0,1
	CRCT	1		CRCT	1
	ARMZ	0,3		ARMZ	0,1
	CRCT	1		DSVS	R00
	ARMZ	0,1	R01:	NADA	
R00:	NADA			CRVL	0,0
	CRVL	0,1		IMPR	
	CRVL	0,0		CRVL	0,1
	CMEG			IMPR	
	DSVF	R01		DMEM	5
	CRVL	0,3		PARA	
	ODILL	0 0			
	CRVL	0,2			
	SOMA	0,2			

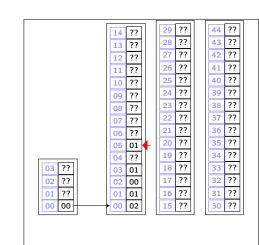


0 00 000000000

INPP			CRVL	0,3
AMEM	2		ARMZ	0,2
AMEM	3			
LEIT			CRVL	0,4
ARMZ	0,0		ARMZ	0,3
CRCT	0			
ARMZ	0,2		CRVL	0,1
CRCT	1		CRCT	1
ARMZ	0,3		ARMZ	0,1
CRCT	1		DSVS	R00
ARMZ	0,1	R01:	NADA	
ROO:NADA			CRVL	0,0
CRVL	0,1		IMPR	
CRVL	0,0		CRVL	0,1
CMEG			IMPR	
DSVF	R01		DMEM	5
CRVL	0,3		PARA	
CRVL	0,2			
SOMA				
ARMZ	0,4			

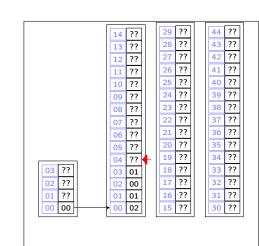


INPP			CRVL	0,3
AMEM	2		ARMZ	0,2
AMEM	3			
LEIT			CRVL	0,4
ARMZ	0,0		ARMZ	0,3
CRCT	0			
ARMZ	0,2		CRVL	0,1
CRCT	1		CRCT	1
ARMZ	0,3		ARMZ	0,1
CRCT	1		DSVS	R00
ARMZ	0,1	R01:	NADA	
ROO:NADA			CRVL	0,0
CRVL	0,1		IMPR	
CRVL	0,0			0,1
CMEG			IMPR	
DSVF	R01		DMEM	5
CRVL	0,3		PARA	
CRVL	0,2			
SOMA				
ARMZ	0,4			

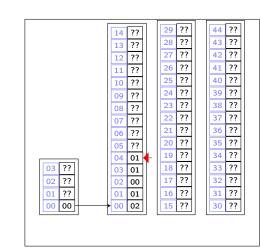


0 000000000 000000000 Exec.Pascal

INPP			CRVL	0,3
AMEM	2		ARMZ	0,2
AMEM	3			
LEIT			CRVL	0,4
ARMZ	0,0		ARMZ	0,3
CRCT	0			
ARMZ	0,2		CRVL	0,1
CRCT	1		CRCT	1
ARMZ	0,3		ARMZ	0,1
CRCT	1		DSVS	R00
ARMZ	0,1	R01:	NADA	
ROO:NADA			CRVL	0,0
CRVL	0,1		IMPR	
CRVL	0,0		CRVL	0,1
CMEG			IMPR	
DSVF	R01		DMEM	5
CRVL	0,3		PARA	
CRVL	0,2			
SOMA				



INPP			CRVL	0,3
AMEM	2		ARMZ	0,2
AMEM	3			
LEIT			${\tt CRVL}$	0,4
ARMZ	0,0		ARMZ	0,3
CRCT	0			
ARMZ	0,2		${\tt CRVL}$	0,1
CRCT	1		CRCT	1
ARMZ	0,3		ARMZ	0,1
CRCT	1		DSVS	R00
ARMZ	0,1	R01:	NADA	
ROO:NADA			${\tt CRVL}$	0,0
CRVL	0,1		${\tt IMPR}$	
CRVL	0,0		CRVL	0,1
CMEG			${\tt IMPR}$	
DSVF	R01		DMEM	5
CRVL	0 0		PARA	
	0,3		I nim	
CRVL			r nith	
CRVL SOMA			1 AILA	

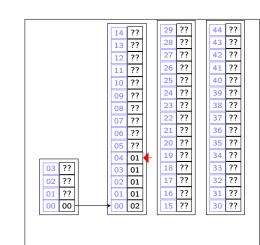


anur o o

Primeira Iteração

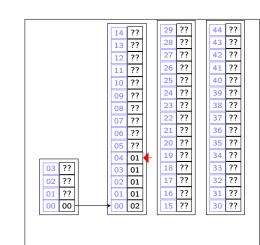
THE

INPP			CRVL	0,3
AMEM	2		ARMZ	0,2
AMEM	3			
LEIT			CRVL	0,4
ARMZ	0,0		ARMZ	0,3
CRCT	0			
ARMZ	0,2		CRVL	0,1
CRCT	1		CRCT	1
ARMZ	0,3		ARMZ	0,1
CRCT	1		DSVS	R00
ARMZ	0,1	R01:	NADA	
ROO:NADA			CRVL	0,0
CRVL	0,1		IMPR	
CRVL	0,0		CRVL	0,1
CMEG			IMPR	
DSVF	R01		DMEM	5
CRVL	0,3		PARA	
CRVL	0,2			
SOMA				
ARMZ	0.4			



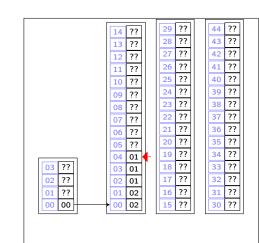
Traducão

```
INPP
                    CRVL 0,3
    AMEM 2
                    ARMZ 0.2
    AMEM 3
    LEIT
                    CRVL 0,4
    ARMZ 0.0
                    ARMZ 0.3
    CRCT 0
    ARMZ 0,2
                    CRVL 0,1
    CRCT 1
                    CRCT 1
    ARMZ 0.3
                    ARMZ 0.1
    CRCT 1
                    DSVS ROO
    ARMZ 0,1
              RO1: NADA
ROO: NADA
                    CRVL 0.0
    CRVL 0,1
                    IMPR
    CRVL 0,0
                    CRVL 0,1
    CMEG
                    IMPR
                    DMEM 5
    DSVF R01
    CRVL 0,3
                    PARA
    CRVL 0.2
    SOMA
```



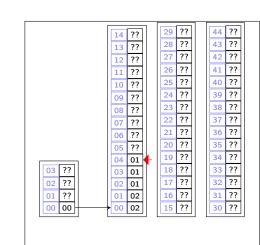
ARMZ 0,4

INPP			CRVL	0,3
AMEM	2		ARMZ	0,2
AMEM	3			
LEIT			CRVL	0,4
ARMZ	0,0		ARMZ	0,3
CRCT	0			
ARMZ	0,2		CRVL	0,1
CRCT	1		CRCT	1
ARMZ	0,3		ARMZ	0,1
CRCT	1		DSVS	R00
ARMZ	0,1	R01:	NADA	
ROO:NADA			CRVL	0,0
CRVL	0,1		IMPR	
CRVL	0,0		CRVL	0,1
CMEG			IMPR	
DSVF	R01		DMEM	5
CRVL	0,3		PARA	
CRVL	0,2			
SOMA				
ARMZ				

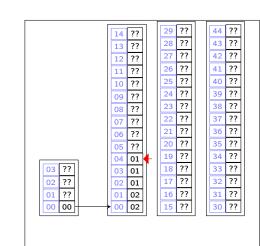


Exec.Pascal Proj

	INPP			CRVL	0,3
	AMEM	2		ARMZ	0,2
	AMEM	3			
	LEIT			CRVL	0,4
	ARMZ	0,0		ARMZ	0,3
	CRCT	0			
	ARMZ	0,2		CRVL	0,1
	CRCT	1		CRCT	1
	ARMZ	0,3		ARMZ	0,1
	CRCT	1		DSVS	R00
	ARMZ	0,1	R01:	NADA	
R00:	NADA			CRVL	0,0
	CRVL	0,1		IMPR	
	CRVL	0,0		CRVL	0,1
	CMEG			IMPR	
	DSVF	R01		DMEM	5
	CRVL	0,3		PARA	
	CRVL	0,2			
	SOMA				
	ARMZ	0,4			

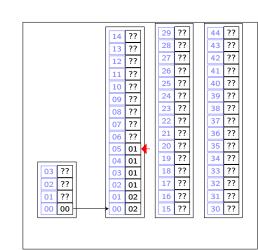


INPP			CRVL	0,3
AMEM	2		ARMZ	0,2
AMEM	3			
LEIT			CRVL	0,4
ARMZ	0,0		ARMZ	0,3
CRCT	0			
ARMZ	0,2		CRVL	0,1
CRCT	1		CRCT	1
ARMZ	0,3		ARMZ	0,1
CRCT	1		DSVS	R00
ARMZ	0,1	R01:	NADA	
ROO:NADA			CRVL	0,0
CRVL	0,1		IMPR	
CRVL	0,0		CRVL	0,1
CMEG			IMPR	
DSVF	R01		DMEM	5
CRVL	0,3		PARA	
CRVL	0,2			
SOMA				
ARMZ	0,4			



Sintaxe

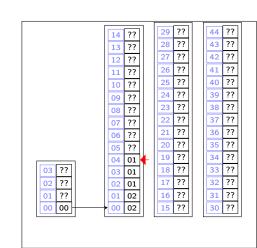
```
INPP
                    CRVL 0,3
    AMEM 2
                    ARMZ 0.2
    AMEM 3
    LEIT
                    CRVL 0,4
    ARMZ 0.0
                    ARMZ 0.3
    CRCT 0
    ARMZ 0,2
                    CRVL 0,1
                    CRCT 1
    CRCT 1
    ARMZ 0.3
                    ARMZ 0.1
    CRCT 1
                    DSVS ROO
    ARMZ 0,1
              RO1: NADA
ROO: NADA
                    CRVL 0.0
    CRVL 0,1
                    IMPR
    CRVL 0,0
                    CRVL 0,1
    CMEG
                    IMPR
                    DMEM 5
    DSVF R01
    CRVL 0,3
                    PARA
    CRVL 0.2
    SOMA
    ARMZ 0,4
```



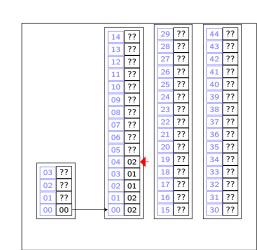
Exec.Pascal

Projeto 000

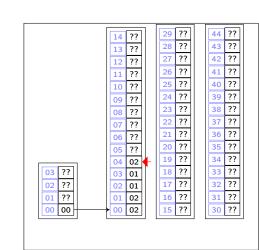
INPP			CRVL	0,3
AMEM	2		ARMZ	0,2
AMEM	3			
LEIT			CRVL	0,4
ARMZ	0,0		ARMZ	0,3
CRCT	0			
ARMZ	0,2		CRVL	0,1
CRCT	1		CRCT	1
ARMZ	0,3		ARMZ	0,1
CRCT	1		DSVS	R00
ARMZ	0,1	R01:	NADA	
ROO:NADA			CRVL	0,0
CRVL	0,1		IMPR	
CRVL	0,0		CRVL	0,1
CMEG			IMPR	
DSVF	R01		DMEM	5
CRVL	0,3		PARA	
CRVL	0,2			
SOMA				
ARMZ	0,4			



INPP			CRVL	0,3
AMEM	2		ARMZ	0,2
AMEM	3			
LEIT			CRVL	0,4
ARMZ	0,0		ARMZ	0,3
CRCT	0			
ARMZ	0,2		CRVL	0,1
CRCT	1		CRCT	1
ARMZ	0,3		ARMZ	0,1
CRCT	1		DSVS	R00
ARMZ	0,1	R01:	NADA	
ROO:NADA			CRVL	0,0
CRVL	0,1		IMPR	
CRVL	0,0		CRVL	0,1
CMEG			${\tt IMPR}$	
DSVF	R01		DMEM	5
CRVL	0,3		PARA	
CRVL	0,2			
SOMA				

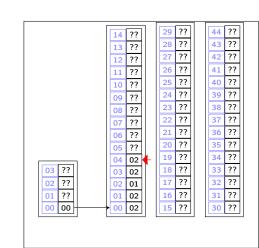


INPP			CRVL	0,3
AMEM	2		ARMZ	0,2
AMEM	3			
LEIT			CRVL	0,4
ARMZ	0,0		ARMZ	0,3
CRCT	0			
ARMZ	0,2		CRVL	0,1
CRCT	1		CRCT	1
ARMZ	0,3		ARMZ	0,1
CRCT	1		DSVS	R00
ARMZ	0,1	R01:	NADA	
ROO:NADA			CRVL	0,0
CRVL	0,1		IMPR	
CRVL	0,0		CRVL	0,1
CMEG			IMPR	
DSVF	R01		DMEM	5
CRVL	0,3		PARA	
CRVL	0,2			
SOMA				
ARMZ				

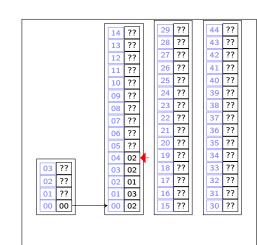


Sintaxe

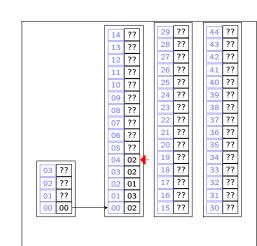
```
INPP
                    CRVL 0,3
    AMEM 2
                    ARMZ 0.2
    AMEM 3
    LEIT
                    CRVL 0,4
    ARMZ 0.0
                    ARMZ 0.3
    CRCT 0
    ARMZ 0,2
                    CRVL 0,1
    CRCT 1
                    CRCT 1
    ARMZ 0.3
                    ARMZ 0.1
    CRCT 1
                    DSVS ROO
    ARMZ 0,1
              RO1: NADA
ROO: NADA
                    CRVL 0.0
    CRVL 0,1
                    IMPR
    CRVL 0,0
                    CRVL 0,1
    CMEG
                    IMPR
                    DMEM 5
    DSVF R01
    CRVL 0,3
                    PARA
    CRVL 0.2
    SOMA
    ARMZ 0,4
```



INPP			CRVL	0,3
AMEM	2		ARMZ	0,2
AMEM	3			
LEIT			CRVL	0,4
ARMZ	0,0		ARMZ	0,3
CRCT	0			
ARMZ	0,2		CRVL	0,1
CRCT	1		CRCT	1
ARMZ	0,3		ARMZ	0,1
CRCT	1		DSVS	R00
ARMZ	0,1	R01:	NADA	
ROO:NADA			CRVL	0,0
CRVL	0,1		IMPR	
CRVL	0,0		CRVL	0,1
CMEG			IMPR	
DSVF	R01		DMEM	5
CRVL	0,3		PARA	
CRVL	0,2			
SOMA				



INPP			CRVL	0,3
AMEM	2		ARMZ	0,2
AMEM	3			
LEIT			CRVL	0,4
ARMZ	0,0		ARMZ	0,3
CRCT	0			
ARMZ	0,2		CRVL	0,1
CRCT	1		CRCT	1
ARMZ	0,3		ARMZ	0,1
CRCT	1		DSVS	R00
ARMZ	0,1	R01:	NADA	
ROO:NADA			CRVL	0,0
CRVL	0,1		IMPR	
CRVL	0,0		CRVL	0,1
CMEG			IMPR	
DSVF	R01		DMEM	5
CRVL	0,3		PARA	
CRVL	0,2			
SOMA				



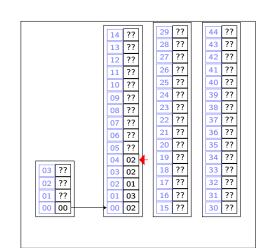
Tradução 0 00 0000000 0000000 Exec.Pascal

Projeto 000 000

Terceira Iteração

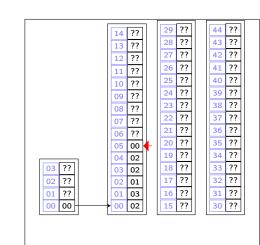
Sintaxe

```
INPP
                    CRVL 0,3
    AMEM 2
                    ARMZ 0.2
    AMEM 3
    LEIT
                    CRVL 0,4
    ARMZ 0.0
                    ARMZ 0.3
    CRCT 0
    ARMZ 0,2
                    CRVL 0,1
    CRCT 1
                    CRCT 1
    ARMZ 0.3
                    ARMZ 0.1
    CRCT 1
                    DSVS ROO
    ARMZ 0,1
              RO1: NADA
ROO: NADA
                    CRVL 0.0
    CRVL 0,1
                    IMPR
    CRVL 0,0
                    CRVL 0,1
    CMEG
                    IMPR
                    DMEM 5
    DSVF R01
    CRVL 0,3
                    PARA
    CRVL 0.2
    SOMA
    ARMZ 0,4
```



Sintaxe

```
INPP
                    CRVL 0,3
    AMEM 2
                    ARMZ 0.2
    AMEM 3
    LEIT
                    CRVL 0,4
    ARMZ 0.0
                    ARMZ 0.3
    CRCT 0
    ARMZ 0,2
                    CRVL 0,1
    CRCT 1
                    CRCT 1
    ARMZ 0.3
                    ARMZ 0.1
    CRCT 1
                    DSVS ROO
    ARMZ 0,1
              RO1: NADA
ROO: NADA
                    CRVL 0.0
    CRVL 0,1
                    IMPR
    CRVL 0,0
                    CRVL 0,1
    CMEG
                    IMPR
                    DMEM 5
    DSVF R01
    CRVL 0,3
                    PARA
    CRVL 0.2
    SOMA
```

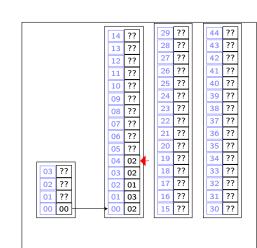


ARMZ 0,4

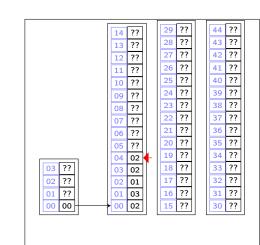
0000000000

Exec.Pascal Proj

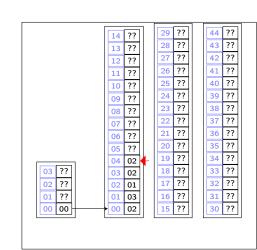
INPP			CRVL	0,3
AMEM	2		ARMZ	0,2
AMEM	3			
LEIT			CRVL	0,4
ARMZ	0,0		ARMZ	0,3
CRCT	0			
ARMZ	0,2		CRVL	0,1
CRCT	1		CRCT	1
ARMZ	0,3		ARMZ	0,1
CRCT	1		DSVS	R00
ARMZ	0,1	R01:	NADA	
ROO:NADA			CRVL	0,0
CRVL	0,1		IMPR	
CRVL	0,0		CRVL	0,1
CMEG			IMPR	
DSVF	R01		DMEM	5
CRVL	0,3		PARA	
CRVL	0,2			
SOMA				
ARMZ	0 4			



INPP			CRVL	0,3
AMEM	2		ARMZ	0,2
AMEM	3			
LEIT			${\tt CRVL}$	0,4
ARMZ	0,0		ARMZ	0,3
CRCT	0			
ARMZ	0,2		CRVL	0,1
CRCT	1		CRCT	1
ARMZ	0,3		ARMZ	0,1
CRCT	1		DSVS	R00
ARMZ	0,1	R01:	NADA	
ROO:NADA			CRVL	0,0
CRVL	0,1		IMPR	
CRVL	0,0		CRVL	0,1
CMEG			IMPR	
DSVF	R01		DMEM	5
CRVL	0,3		PARA	
CRVL	0,2			
SOMA				
ARMZ	0,4			

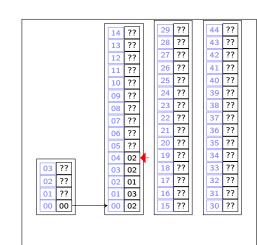


INPP			CRVL	0,3
AMEM	2		ARMZ	0,2
AMEM	3			
LEIT			CRVL	0,4
ARMZ	0,0		ARMZ	0,3
CRCT	0			
ARMZ	0,2		CRVL	0,1
CRCT	1		CRCT	1
ARMZ	0,3		ARMZ	0,1
CRCT	1		DSVS	R00
ARMZ	0,1	R01:	NADA	
ROO:NADA			CRVL	0,0
CRVL	0,1		IMPR	
CRVL	0,0		CRVL	0,1
CMEG			IMPR	
DSVF	R01		DMEM	5
CRVL	0,3		PARA	
CRVL	0,2			
SOMA				

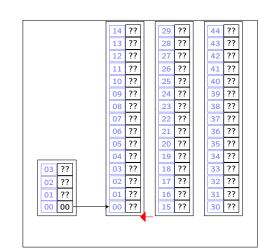


Tradução 0 00 0000000000 0000000000

```
INPP
                    CRVL 0,3
    AMEM 2
                    ARMZ 0.2
    AMEM 3
    LEIT
                    CRVL 0,4
    ARMZ 0.0
                    ARMZ 0.3
    CRCT 0
    ARMZ 0,2
                    CRVL 0,1
    CRCT 1
                    CRCT 1
    ARMZ 0.3
                    ARMZ 0.1
    CRCT 1
                    DSVS ROO
    ARMZ 0,1
              RO1: NADA
ROO: NADA
                    CRVL 0.0
    CRVL 0,1
                    IMPR
    CRVL 0,0
                    CRVL 0,1
    CMEG
                    IMPR
                    DMEM 5
    DSVF R01
    CRVL 0,3
                    PARA
    CRVL 0.2
    SOMA
    ARMZ 0,4
```



INPP			CRVL	0,3
AMEM	2		ARMZ	0,2
AMEM	3			
LEIT			CRVL	0,4
ARMZ	0,0		ARMZ	0,3
CRCT	0			
ARMZ	0,2		CRVL	0,1
CRCT	1		CRCT	1
ARMZ	0,3		ARMZ	0,1
CRCT	1		DSVS	R00
ARMZ	0,1	R01:	NADA	
ROO:NADA			CRVL	0,0
CRVL	0,1		IMPR	
CRVL	0,0		CRVL	0,1
CMEG			TMPR.	
			THER	
DSVF	R01		DMEM	5
DSVF CRVL				-
	0,3		DMEM	-
CRVL	0,3		DMEM	-



Exec.Pascal

??

??

Terceira Iteração

INPP			CRVL	0,3
AMEM	2		ARMZ	0,2
AMEM	3			
LEIT			CRVL	0,4
ARMZ	0,0		ARMZ	0,3
CRCT	0			
ARMZ	0,2		CRVL	0,1
CRCT	1		CRCT	1
ARMZ	0,3		ARMZ	0,1
CRCT	1		DSVS	R00
ARMZ	0,1	R01:	NADA	
ROO:NADA			CRVL	0,0
CRVL	0,1		IMPR	
CRVL	0,0		CRVL	0,1
CMEG			IMPR	
DSVF	R01		DMEM	5
CRVL	0,3		PARA	
CRVL	0,2			
SOMA				

14 29 44 ?? ?? ?? 28 43 ?? ?? ?? 42 26 41 ?? ?? ?? 10 25 40 09 ?? 24 ?? 39 ?? ?? ?? ?? 08 23 38 ?? ?? ?? 37 ?? ?? ?? 06 21 36 ?? ?? 20 ?? ?? ?? ?? 04 19 34 ?? ?? 03 03 ?? 18 33 ?? ?? ?? ?? 02 17 ?? ?? 16 ?? 31 00 ?? 00 ?? ?? 30 ??

ARMZ 0,4

Exec.Pascal

Sintaxe

- A execução das instruções MEPA uma a uma apresenta vários problemas didáticos:
 - É muito detalhada, podendo causar erros.
 - O nível de detalhe esconde muitas informações pertinentes, como por exemplo: Qual é a variável que reside no endereço léxico 0.0?
- Por estas e outras razões fiz a opção de simular a execução do programa Pascal diretamente na máquina virtual.
- Isto aumenta o nível de abstração e permite explicar o funcionamento conceitual do ambiente de execução.
- Se existe alguma coisa positiva com a execução anterior, acredito que seja saber como se sente um computador.



Traducão

44

43

42

41 ??

40

39 ??

38

37

36

34

33

31 ??

30 ??

??

??

??

??

??

??

??

??

??

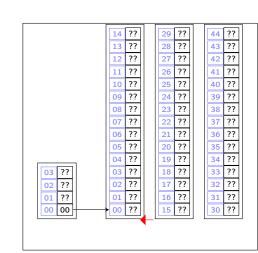
??

Execução

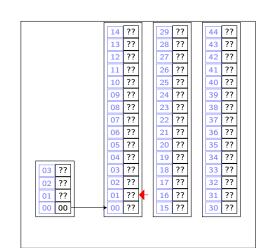
```
program comandoWhile;
var n. k: integer:
    f1, f2, f3:integer:
begin
   read(n)
   f1:=0: f2:=1: k:=1:
   while (k <= n) do
   begin
      f3:=f2+f1:
      f1:=f2;
      f2:=f3;
      k := k+1
   end:
   write(n.k)
   end.
```

14 29 ?? ?? 28 ?? ?? 26 ?? ?? 10 25 09 ?? 24 ?? ?? ?? 08 23 ?? ?? ?? ?? 06 21 ?? ?? ?? ?? 04 19 ?? 03 ?? 18 ?? ?? ?? 17 ?? ?? ?? 16 00 ?? ?? 00 00

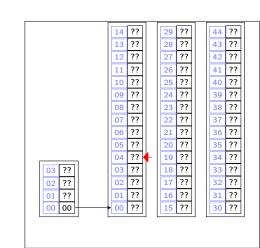
```
program comandoWhile;
var n. k: integer:
    f1, f2, f3:integer:
begin
   read(n)
   f1:=0: f2:=1: k:=1:
   while (k <= n) do
   begin
      f3:=f2+f1:
      f1:=f2;
      f2:=f3;
      k := k+1
   end:
   write(n.k)
   end.
```



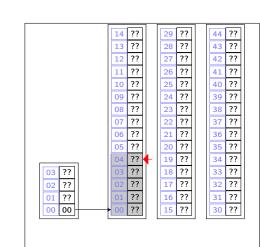
```
program comandoWhile;
var n. k: integer:
    f1, f2, f3:integer:
begin
   read(n)
   f1:=0: f2:=1: k:=1:
   while (k <= n) do
   begin
      f3:=f2+f1:
      f1:=f2;
      f2:=f3;
      k := k+1
   end:
   write(n.k)
   end.
```



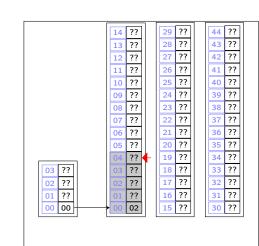
```
program comandoWhile;
var n. k: integer:
    f1, f2, f3:integer:
begin
   read(n)
   f1:=0: f2:=1: k:=1:
   while (k <= n) do
   begin
      f3:=f2+f1:
      f1:=f2;
      f2:=f3;
      k := k+1
   end:
   write(n.k)
   end.
```



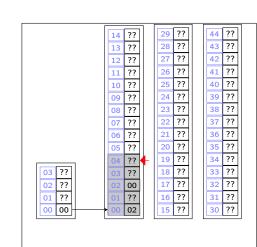
```
program comandoWhile;
var n. k: integer:
    f1, f2, f3:integer:
begin
   read(n)
   f1:=0: f2:=1: k:=1:
   while (k <= n) do
   begin
      f3:=f2+f1:
      f1:=f2;
      f2:=f3;
      k := k+1
   end:
   write(n.k)
   end.
f3(0,4)
f2(0.3)
f1(0.2)
k (0,1)
n (0,0)
```



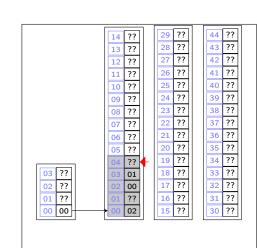
```
program comandoWhile;
var n. k: integer:
    f1, f2, f3:integer:
begin
   read(n)
   f1:=0: f2:=1: k:=1:
   while (k <= n) do
   begin
      f3:=f2+f1:
      f1:=f2;
      f2:=f3;
      k := k+1
   end:
   write(n.k)
   end.
f3(0,4)
f2(0.3)
f1(0.2)
k (0,1)
n (0,0)
```



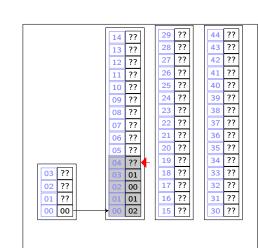
```
program comandoWhile;
var n. k: integer:
    f1, f2, f3:integer:
begin
   read(n)
   f1:=0: f2:=1: k:=1:
   while (k <= n) do
   begin
      f3:=f2+f1:
      f1:=f2;
      f2:=f3;
      k := k+1
   end:
   write(n.k)
   end.
f3(0,4)
f2(0.3)
f1(0.2)
k (0,1)
n (0,0)
```



```
program comandoWhile;
var n. k: integer:
    f1, f2, f3:integer:
begin
   read(n)
   f1:=0: f2:=1: k:=1:
   while (k <= n) do
   begin
      f3:=f2+f1:
      f1:=f2;
      f2:=f3;
      k := k+1
   end:
   write(n.k)
   end.
f3(0,4)
f2(0.3)
f1(0.2)
k (0,1)
n (0,0)
```



```
program comandoWhile;
var n. k: integer:
    f1, f2, f3:integer:
begin
   read(n)
   f1:=0: f2:=1: k:=1:
   while (k <= n) do
   begin
      f3:=f2+f1:
      f1:=f2;
      f2:=f3;
      k := k+1
   end:
   write(n.k)
   end.
f3(0,4)
f2(0.3)
f1(0.2)
k (0,1)
n (0,0)
```

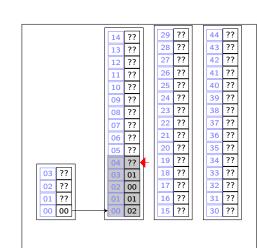


Traducão

Exec Pascal

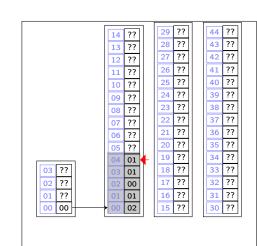
00000000000000

```
program comandoWhile;
var n. k: integer:
    f1, f2, f3:integer:
begin
   read(n)
   f1:=0: f2:=1: k:=1:
   while (k <= n) do
   begin
      f3:=f2+f1:
      f1:=f2;
      f2:=f3;
      k := k+1
   end:
   write(n.k)
   end.
f3(0,4)
f2(0.3)
f1(0.2)
k (0,1)
n (0,0)
```



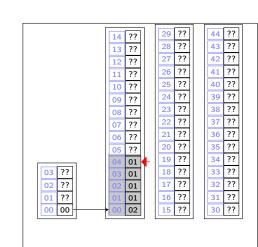
Traducão

```
program comandoWhile;
var n. k: integer:
    f1, f2, f3:integer:
begin
   read(n)
   f1:=0: f2:=1: k:=1:
   while (k <= n) do
   begin
      f3:=f2+f1:
      f1:=f2;
      f2:=f3;
      k := k+1
   end:
   write(n.k)
   end.
f3(0,4)
f2(0.3)
f1(0.2)
k (0,1)
n (0,0)
```



Traducão

```
program comandoWhile;
var n. k: integer:
    f1, f2, f3:integer:
begin
   read(n)
   f1:=0: f2:=1: k:=1:
   while (k <= n) do
   begin
      f3:=f2+f1:
      f1:=f2;
      f2:=f3;
      k := k+1
   end:
   write(n.k)
   end.
f3(0,4)
f2(0.3)
f1(0.2)
k (0,1)
n (0,0)
```

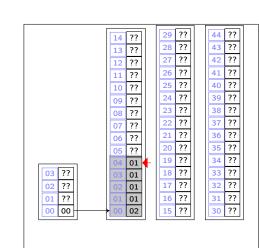


Tradução 0 00 0000000000 0000000000

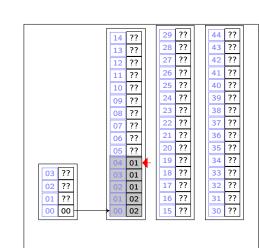
000000000 0000000 0000000 0000000 Exec.Pascal
0
00000000000000000

00000000

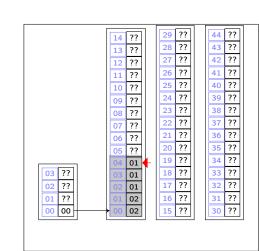
```
program comandoWhile;
var n. k: integer:
    f1, f2, f3:integer:
begin
   read(n)
   f1:=0: f2:=1: k:=1:
   while (k <= n) do
   begin
      f3:=f2+f1:
      f1:=f2;
      f2:=f3;
      k := k+1
   end:
   write(n.k)
   end.
f3(0,4)
f2(0.3)
f1(0.2)
k (0,1)
n (0,0)
```



```
program comandoWhile;
var n. k: integer:
    f1, f2, f3:integer:
begin
   read(n)
   f1:=0: f2:=1: k:=1:
   while (k <= n) do
   begin
      f3:=f2+f1:
      f1:=f2;
      f2:=f3;
      k := k+1
   end:
   write(n.k)
   end.
f3(0,4)
f2(0.3)
f1(0.2)
k (0,1)
n (0,0)
```

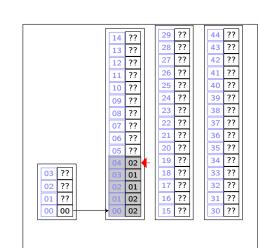


```
program comandoWhile;
var n. k: integer:
    f1, f2, f3:integer:
begin
   read(n)
   f1:=0: f2:=1: k:=1:
   while (k <= n) do
   begin
      f3:=f2+f1:
      f1:=f2;
      f2:=f3;
      k := k+1
   end:
   write(n.k)
   end.
f3(0,4)
f2(0.3)
f1(0.2)
k (0,1)
n (0,0)
```

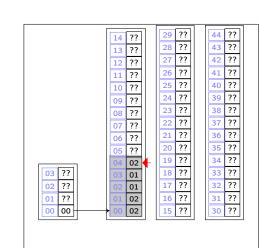


Execução

```
program comandoWhile;
var n. k: integer:
    f1, f2, f3:integer:
begin
   read(n)
   f1:=0: f2:=1: k:=1:
   while (k <= n) do
   begin
      f3:=f2+f1:
      f1:=f2;
      f2:=f3;
      k := k+1
   end:
   write(n.k)
   end.
f3(0,4)
f2(0.3)
f1(0.2)
k (0,1)
n (0,0)
```

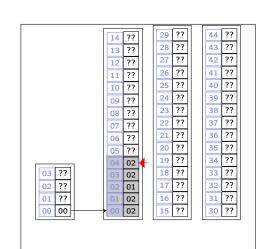


```
program comandoWhile;
var n. k: integer:
    f1, f2, f3:integer:
begin
   read(n)
   f1:=0: f2:=1: k:=1:
   while (k <= n) do
   begin
      f3:=f2+f1:
      f1:=f2;
      f2:=f3;
      k := k+1
   end:
   write(n.k)
   end.
f3(0,4)
f2(0.3)
f1(0.2)
k (0,1)
n (0,0)
```



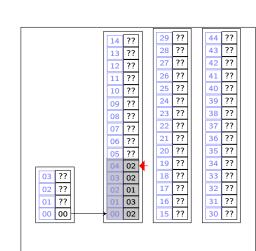
Execução

```
program comandoWhile;
var n. k: integer:
    f1, f2, f3:integer:
begin
   read(n)
   f1:=0: f2:=1: k:=1:
   while (k <= n) do
   begin
      f3:=f2+f1:
      f1:=f2;
      f2:=f3;
      k := k+1
   end:
   write(n.k)
   end.
f3(0,4)
f2(0.3)
f1(0.2)
k (0,1)
n (0,0)
```



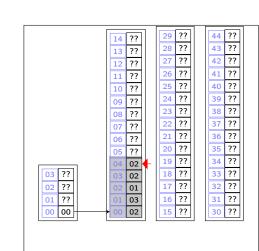
Execução

```
program comandoWhile;
var n. k: integer:
    f1, f2, f3:integer:
begin
   read(n)
   f1:=0: f2:=1: k:=1:
   while (k <= n) do
   begin
      f3:=f2+f1:
      f1:=f2;
      f2:=f3;
      k := k+1
   end:
   write(n.k)
   end.
f3(0,4)
f2(0.3)
f1(0.2)
k (0,1)
n (0,0)
```

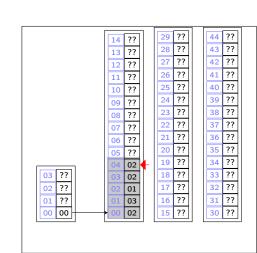


•00

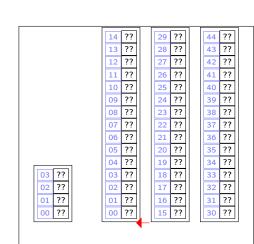
```
program comandoWhile;
var n. k: integer:
    f1, f2, f3:integer:
begin
   read(n)
   f1:=0: f2:=1: k:=1:
   while (k <= n) do
   begin
      f3:=f2+f1:
      f1:=f2;
      f2:=f3;
      k := k+1
   end:
   write(n.k)
   end.
f3(0,4)
f2(0.3)
f1(0.2)
k (0,1)
n (0,0)
```



```
program comandoWhile;
var n. k: integer:
    f1, f2, f3:integer:
begin
   read(n)
   f1:=0: f2:=1: k:=1:
   while (k <= n) do
   begin
      f3:=f2+f1:
      f1:=f2;
      f2:=f3;
      k := k+1
   end;
   write(n.k)
   end.
f3(0,4)
f2(0.3)
f1(0.2)
k (0,1)
n (0,0)
```



```
program comandoWhile;
var n, k: integer;
    f1, f2, f3:integer;
begin
    read(n)
    f1:=0; f2:=1; k:=1;
while (k <= n) do
    begin
        f3:=f2+f1;
        f1:=f2;
        f2:=f3;
        k:=k+1
    end;
    write(n,k)
end</pre>
```



Projeto

- Implemente o comando while no compilador.
 - Baseie-se no esquema de tradução.
 - Pense em como contornar o problema de vários comandos while aninhados. Os diversos rótulos de entrada (ROO) e de saída (R01) não devem ser confundidos.
 - Sugestões:
 - Criar uma subrotina "geradora" do próximo rótulo.
 - Criar uma pilha de rótulos.
 - No início do while, empilhar os dois rótulos.
 - Ao final, desempilhá-los.



Página para anotações

Licença

- Slides desenvolvidos somente com software livre:
 - LATEX usando beamer;
 - Inkscape.
- Licença:
 - Creative Commons Atribuição-Uso Não-Comercial-Vedada a Criação de Obras Derivadas 2.5 Brasil License. http: //creativecommons.org/licenses/by-nc-nd/2.5/br/