

Lista de Exercícios No. 3

1 - Considere uma máquina com memória de 64 K endereços que armazenam 8 bits. Os 2000 bytes iniciais são reservados para o sistema. O compilador usa estratégia de alocação de memória de pilha para registros de ativação e alocação em heap para dados alocados dinamicamente. deseja-se executar o seguinte programa:

Prog Memoria

Var i,j:inteiro;

Proc FRec;

Var k:inteiro;

x:^inteiro;

Inicio

k:=3; i:=i-1; x:=@k;

Escreva(tela,"x=",x,"end=",@x,"valor apontado=",x^);

Se i>0 entao FRec;

New(x);

x^:=0;

Escreva(tela,"topo pilha=",topo_pilha);

Escreva("x=",x,"end=",@x,"valor apontado=",x^);

Fim;

Inicio

i:=2; j:=0;

Escreva(tela,"topo pilha=", topo_pilha);

FRec;

Escreva(tela,"topo pilha=", topo_pilha);

Escreva(tela,"i=",i,"end=",@i);

Escreva(tela,"topo heap=",topo_heap);

Fim.

Considere inteiros ocupando 2 bytes, tamanho do programa principal de 1000 bytes e tamanho de FRec de 500 bytes. O programa gasta 12 bytes com temporários e FRec 22. FRec ocupa 10 bytes para status. *topo-pilha* e *topo-heap* retornam a primeira posição de memória vazia para pilha e heap, respectivamente. Execute o programa, construindo o mapa de memória.

2 - Diga o que será impresso pelo seguinte programa, considerando (a) regra de escopo léxico e (b) regra de escopo dinâmico:

```
Prog principal;
| Var a,b,c,d:inteiro;
|
| Proc X;
| | Var b,d:inteiro;
| Inicio
| | b:=a; a:=a+2;
| | d:=1; c:=d*a+1;
| | Escreva(a,b,c,d);
| Fim;
|
| Proc Y;
| | Var a,d:inteiro;
| |
| | Proc X;
| | | var d:inteiro;
| | Inicio
| | | d:=1; a:=b+c;
| | | Escreva(a,b,c,d);
| | Fim
| Inicio
| | a:=1; a:=c*a; d:=2;
| | X;
| | Escreva(a,b,c,d);
| Fim;
|
| Proc Z;
| | Var a,b,c:inteiro;
| Inicio
| | a:=1; b:=0; c:=a; d:=1;
| | Y;
| | Escreva(a,b,c,d);
| Fim;
|
| Inicio
| | a:=0; b:=1; c:=2;
| | Z; X; Y;
| Fim.
```

3 - Dado o programa abaixo, diga o que será escrito se a passagem de parâmetros for feita (a) por valor e (b) por referência:

```
Prog principal;  
|  Var a,b,c:inteiro;  
|  
|  Proc inic(b,c,d);  
|  Inicio  
|  |  d:=3; c:=b; b:=6; a:=a+d;  
|  Fim;  
|  
|  Proc P(b,c,d);  
|  Inicio  
|  |  a:=a+d; c:=d*b; d:=c+b-1;  
|  Fim;  
|  
Inicio  
|  a:=2; b:=1; c:=0;  
|  inic(a,b,c);  
|  P(c,b,a);  
|  Escreva(a,b,c);  
Fim.
```

4 - Dado o programa abaixo, diga o que será escrito considerando:

- (a) passagem por valor e escopo dinâmico
- (b) passagem por referência e escopo dinâmico
- (c) passagem por valor e escopo léxico
- (d) passagem por referência e escopo léxico

Prog Principal;

```
| Var x,y,z:byte;  
|   p,q:*byte;
```

```
| Proc A(y:byte);  
| Inicio  
|   x:=y+z; y:=4;  
| Fim;
```

```
| Proc B(x:byte; z:*byte);  
| Inicio  
|   *z:=5;  
|   New(p);  
|   *p:=x;  
| Fim;
```

```
| Proc C(q:*byte);  
|   Var r:*byte;  
|   z:byte;  
| Inicio  
|   z:=3;  
|   A(x);  
|   New(r);  
|   *r:=x;  
|   Free(q);  
|   p:=r;  
| Fim;
```

Inicio

```
| x:=0; y:=1; z:=2;  
| p:=&x; q:=&y;  
| B(y,q);  
| C(p);  
| /* Mostre aqui o mapa de memoria */  
| Escreva(x,y,z,p,q,*p,*q);
```

Fim.

5 - Dado os códigos intermediários abaixo, proceda a alocação de registradores para os temporários. Após a alocação, qual seria o código ASSEMBLY 8086 gerado? Considere registradores de 8 bits.

a)

```
1:      t6 = 0;
2:      t1 = t7;
3:      t2 = t6 * 4;
4:      t3 = t1 + t2;
5: L1:  if (t6 > 100) goto L2
6:      t3 = t1 + 1;
7:      t6 = t6 + 1;
8:      t2 = t6 * 4;
9:      t5 = t7;
10:     if (t3 <= t7) goto L3
11:     t3 = t5 - t2;
12:     t1 = t5;
13:     goto L2
14: L3:  t4 = t7;
15:     t3 = t4 + t2;
16:     t1 = t4;
17: L2:
```

b) Para este exercício, i e x são variáveis do tipo byte e *t é o conteúdo apontado por t:

```
01:     t1 = 8
02:     t2 = i + 10
03:     t3 = t1 + t2
04:     t4 = *t3
05:     t5 = t4 * x
06:     t6 = i + 20
07:     t7 = t6 * 8
08:     t8 = *t7
09:     t9 = t5 + t8
10:     *t7 = t9
```