

# Fundamentos de Programação 1

## Linguagem C

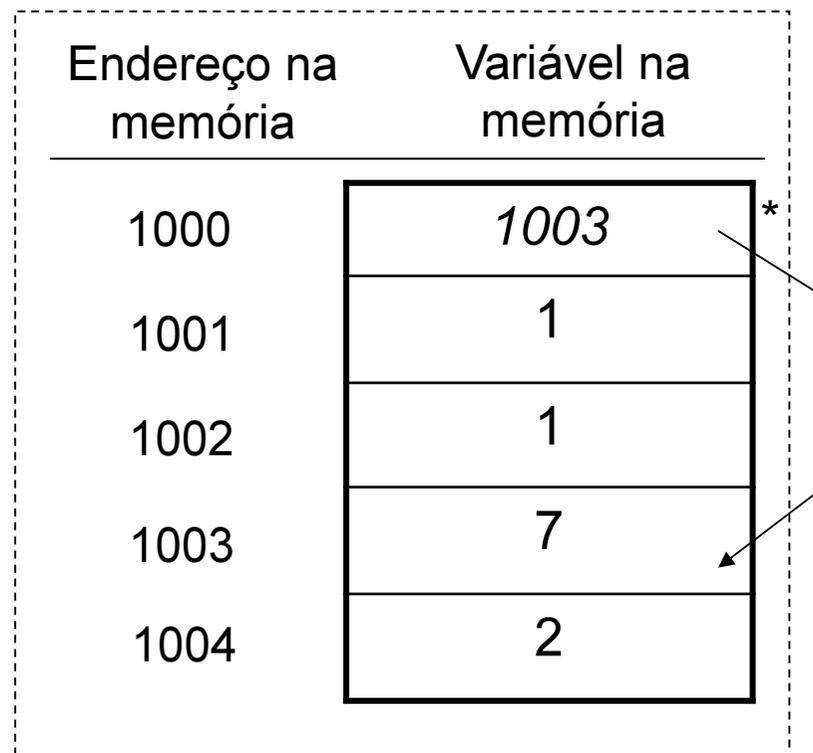
### “Ponteiros”

---

Slice 14

Prof. SIMÃO

Um ponteiro é uma variável que contém um endereço de memória. Este endereço é normalmente a posição de uma outra variável na memória. Se uma variável contém o endereço de um outra, então a primeira variável é dita apontar para a segunda [Schiltd, 1996].



# Exemplo

```
#include <stdlib.h>
#include <stdio.h>

int main()
{
    int x, y;
    x = 1;

    int* p;           // Declaração de um ponteiro para inteiro.
                    // Isto significa que p poderá apontar para um inteiro.
                    // Um ponteiro é declarado com auxílio do *

    p = &x;          // Aqui o ponteiro recebe o endereço da variável x na memória.
                    // Para um ponteiro receber o endereço de uma variável se faz
                    // necessário utilizar o operador & antes da variável.

    y = *p;         // Aqui a variável y recebe o valor da variável apontada pelo
                    // ponteiro p. Na prática, isto significa que y terá o valor
                    // da variável x.
                    // Para uma variável receber o valor da variável apontado por
                    // um ponteiro, faz-se necessário utilizar o operador *.

                    // Assim sendo, o operador * serve para diversas funções em C:
                    // - serve para declarar um ponteiro.
                    // - serve para informar o valor de uma variável apontada.
                    // - serve para multiplicar dois números (sua função primordial).
                    // Portanto, o uso do * depende do contexto.

    // ...
}
```

```
// ...  
  
printf ("O valor da variável x : %i. \n", x);  
  
printf ("O endereço da variável x : %p. \n", &x);  
printf ("\n");  
  
printf ("O valor da variável x via o ponteiro p : %i. \n", *p);  
  
printf ("O endereço da variável x via o ponteiro p: %p. \n", p);  
  
printf ("O endereço do ponteiro p : %p. \n", &p);  
printf ("\n");  
  
printf ("O valor da variável y, adquirido de x, via p: %i. \n", y);  
  
printf ("O endereço da variável y : %p. \n", &y);  
printf ("\n");  
  
// O %p serve para expressar o valor de um endereço de memória.  
  
system("Pause");  
return 0;  
}
```

# Exemplo

```
#include <stdlib.h>
#include <stdio.h>

int main()
{
    int x, y;
    x = 1;

    int *p;
    p = &x;
    y = *p;

    printf ("O valor da variável x : %i. \n", x);
    printf ("O endereço da variável x : %p. \n", &x);
    printf ("\n");
    printf ("O valor da variável x via o ponteiro p : %i. \n", *p);
    printf ("O endereço da variável x via o ponteiro p: %p. \n", p);
    printf ("O endereço do ponteiro p : %p. \n", &p);
    printf ("\n");
    printf ("O valor da variável y, adquirido de x, via p: %i. \n", y);
    printf ("O endereço da variável y : %p. \n", &y);
    printf ("\n");

    // O %p serve para expressar o valor de um endereço de memória.

    system("Pause");
    return 0;
}
```

```
C:\Simao\Simao2007\Disciplinas Minhas\Computação \Exemplos C\PonteirosExemplo1.exe
0 valor da variavel x : 1.
0 endereco da variavel x : 0022FF74.

0 valor da variavel x via o ponteiro p : 1.
0 endereco da variavel x via o ponteiro p: 0022FF74.
0 endereco do ponteiro p : 0022FF6C.

0 valor da variavel y, adquirido de x, via p: 1.
0 endereco da variavel y : 0022FF70.

Press any key to continue . . . _
```

# Exemplo 2

```
#include <stdlib.h>
#include <stdio.h>
int main()
{   int x, y;   x = 1;

    int *p;
    p = &x;
    y = *p;

    printf ("O valor da variável x : %i. \n", x);
    printf ("O endereço da variável x : %p. \n \n", &x);

    printf ("O valor da variável x via o ponteiro p : %i. \n", *p);
    printf ("O endereço da variável x via o ponteiro p : %p. \n", p);
    printf ("O endereço do ponteiro p : %p. \n \n", &p);

    printf ("O valor da variável y, adquirido de x, via p: %i. \n", y);
    printf ("O endereço da variável y : %p. \n \n", &y);

    *p = 5;           // Aqui muda-se o valor de x via *p.

    printf("O novo valor da variável x : %i. \n", x);
    printf("O endereço da variável x : %p. \n \n", &x);

    printf("O novo valor da variável x via o ponteiro p: %i. \n", *p);
    printf("O endereço da variável x via o ponteiro p: %p. \n", p);
    printf("O endereço do ponteiro p : %p. \n \n", &p);

    ...
}
```

```
C:\Simao\Simao2007\Disciplinas Minhas\Computação \Exemplos C\PonteirosExemplo1B.exe
0 valor da variavel x : 1.
0 endereco da variavel x : 0022FF74.

0 valor da variavel x via o ponteiro p : 1.
0 endereco da variavel x via o ponteiro p: 0022FF74.
0 endereco do ponteiro p : 0022FF6C.

0 valor da variavel y, adquirido de x, via p: 1.
0 endereco da variavel y : 0022FF70.

0 novo valor da variavel x : 5.
0 endereco da variavel x : 0022FF74.

0 novo valor da variavel x via o ponteiro p : 5.
0 endereco da variavel x via o ponteiro p: 0022FF74.
0 endereco do ponteiro p : 0022FF6C.
Press any key to continue . . .
```

# Exemplo 3

---

```
#include <stdlib.h>
#include <stdio.h>
```

```
// O segundo parâmetro é um ponteiro de real (float)
void area_quadrado (float ld, float *ar)
{
    // Aqui o valor da variável apontada pelo ponteiro ar será alterada.

    *ar = (ld * ld);

    // bem por isto que se diz que é uma passagem por referência...
}
```

```
int main()
{
    float lado, area;
    area = -1;
    lado = 2;

    // No segundo parâmetro passa-se o endereço da variável area
    // para o ponteiro ar, definido na função acima (vide acima).
    area_quadrado (lado, &area);

    printf("O valor da área é: %f \n", area);

    system("Pause");
    return 0;
}
```