
Funções Parte III

Ponteiros - Conceitos Básicos
Passagem de Parâmetros por Referência em Funções
Exercício

Ponteiros - Conceitos Básicos

- Uma **variável** é armazenada em um certo número de bytes em uma determinada posição de memória.
- Um **ponteiro** é uma variável que contém o endereço de outra variável.
- Os **ponteiros** são usados para acessar e manipular conteúdos em determinado endereço de memória.

Ponteiros - Conceitos Básicos

- Acesso ao endereço de memória da variável:
 &<nome_var>
 - Declaração de um ponteiro:
 <tipo> * <nome_var_ponteiro>
 - Atribuição de um ponteiro:
 <ponteiro_tipoX> = & <variável_tipoX>
-

Ponteiros - Conceitos Básicos

- Exemplos de declaração de ponteiros:

```
int *p;  
char *q;  
float *r,*s;
```

- Exemplos atribuição de ponteiros:

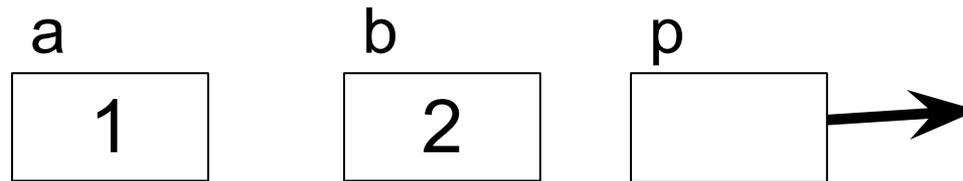
```
p = 0;  
p = NULL; /* equivalente a p = 0 */  
p = &i;  
p = (int *) 1776; /* valor absoluto do endereço*/
```

Ponteiros - Conceitos Básicos

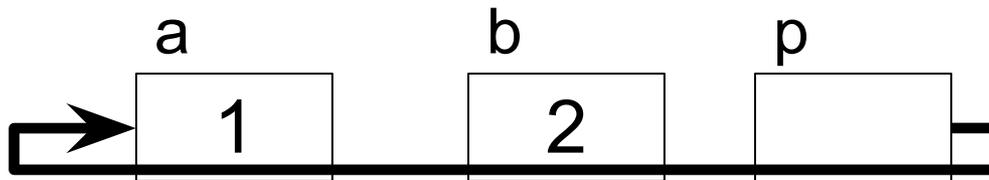
- O operador unário $*$ representa indireção.
 - Se p é um ponteiro, então $*p$ é o valor da variável da qual p é o endereço.
 - O valor direto de p é o endereço de memória.
 - $*p$ é o valor indireto de p , pois representa o valor armazenado no endereço de memória.
-

Ponteiros - Conceitos Básicos

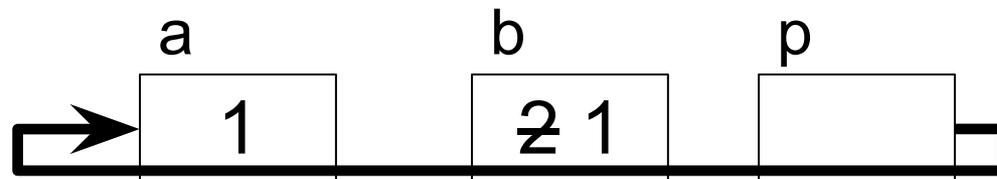
```
int a=1, b=2, *p;
```



```
p=&a
```



```
b = *p; ⇔ b=a;
```



Ponteiros - Conceitos Básicos

Exemplo:

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int i=7, *p = &i;
    printf("%s%d\n%s%p\n", " Valor de i: ", *p,
        "Endereco de i:", p);
    return 0;
}
```

Valor de i: 7 Endereco de i: 0028FF44
--

Passagem de parâmetros por referência em funções

- O endereço de memória da variável é fornecido à função e não uma cópia do valor da variável.
 - Qualquer alteração executada pela função ocorre na posição de memória fornecida.
 - Portanto, as alterações permanecem quando a função é encerrada.
 - Ponteiros, são usados nas passagens por referência.
-

Exemplo:

```
#include <stdio.h>
void swap(int *, int *);
int main(void)
{
    int i=3, j=5;
    swap(&i, &j);
    printf(“%d %d\n”, i, j); /* 5 e 3 são exibidos */
    return 0;
}
void swap (int *p, int *q)
{
    int tmp;
    tmp = *p;
    *p = *q;
    *q = tmp;
}
```

Passagem de parâmetros por referência em funções

- As variáveis `i` e `j` são passadas por referência, ou seja, o endereço de memória das variáveis é repassado à função.

```
swap (&i, &j)
```

- Os ponteiros `*p` e `*q`, declarados no argumento na função `swap`, passam a referenciar a posição de memória das variáveis `i` e `j`.

```
void swap (int *p, int *q)
```

Exercício

Escreva um programa que receba um número inteiro representando a quantidade total de segundos e, usando passagem de parâmetros por referência, converta a quantidade informada de segundos em Horas, Minutos e Segundos. Imprima o resultado da conversão no formato HH:MM:SS. Utilize o seguinte protótipo de função:

```
void converteHora(int total_segundos, int *hora, int *min, int *seg);
```
