INSTITUTO FEDERAL
BAIANO

Estrutura de Dados

Ponteiros

Prof. José Honorato Ferreira Nunes
honoratonunes@softwarelivre.org
http://softwarelivre.org/zenorato

Ponteiro

C permite que o programador referencie a posição de objetos bem como os próprios objetos (isto é, o conteúdo de suas posições).

Por exemplo, se x for declarado como um inteiro, &x se referirá à posição reservada para conter x. &x é chamado ponteiro.

Ponteiro

Um ponteiro é uma variável que contém um endereço de memória;

Assim como um int guarda inteiro, um float guarda ponto flutuante e um char guarda caracteres, o ponteiro guarda um endereço de memória;

Para cada tipo de dado existente, há um tipo ponteiro que pode armazenar endereços de memória onde existem valores do tipo correspondente armazenados.

Ponteiro

Alguns motivos para se usar ponteiros:

- Passagem de parâmetros para uma função por referência.
- Passamos o endereço (um ponteiro) da variável argumento;
- Forma elegante de passar matrizes e strings como argumentos para funções;
- A utilização sensata de ponteiros deixa o programa mais rápido;
- Ponteiros são a base para a criação de estruturas de dados mais avançadas, como listas, pilhas, filas, árvores, etc(Estrutura de Dados I);

Ponteiro

- Para declararmos um ponteiro, usamos a mesma palavra do tipo com os nomes das variáveis precedidas pelo caractere *. Assim, podemos escrever: int *p para declarar um ponteiro do tipo int.
- Podemos usar o endereço unário & ("endereço de"), que resulta no endereço da posição da memória reservada para a variável. O operador unário * ("conteúdo de"), aplicado a variáveis do tipo ponteiro, acessa o conteúdo do endereço de memória armazenado pela variável ponteiro.

Ponteiro

Para trabalharmos com alocação dinâmica, devemos alocar porções de memória utilizando a função ***malloc()***.

```
main(){
    int *ptr, i;
    ptr = (int *) malloc (sizeof(int));
    *ptr = 5800;
    i = *ptr;
    printf("%i",i);
    i = 4200;
    printf("%i",i);
}
```

Ponteiro: exemplo

```
main()
{
    char c, *pc, x;
    c = 'A';
    pc = &c;
    printf("%c", *pc);
    x = *pc;
}
```

c	pc	x
1000	1001	1003
Lixo	Lixo	Lixo

Ponteiro: exemplo

Fazemos a variável ***c*** receber o valor 'A' como segue: ***c = 'A'***

c	pc	x
1000	1001	1003
'A'	Lixo	Lixo

Fazemos a variável ***pc*** receber o endereço da variável ***c*** como segue: **pc = &c**

c	pc	x
1000	1001	1003
'A'	1000	Lixo

x = *pc

c	pc	x
1000	1001	1003
'A'	1000	'A'

Ponteiro: exemplo

```
main()
{
    char c, *pc, x;
    c = 'A';
    pc = &c;
    x = *pc;
    printf("\n%c", c);
    printf("\n%c", *pc);
    printf("\n%c", x);

}
```

Ponteiro para Funções

```
#include<stdio.h>
void troca(int *px, int *py){
    int temp;
    temp = *px;
    *px = *py;
    *py = temp;
}

main (){
    int a = 5, b = 7;
    troca(&a, &b);
    printf("%d %d \n", a, b);
}
```