

Estrutura de Dados

Ponteiros

Wolney Henrique Queiroz Freitas
wolney.freitas@professores.unifavip.edu.br

Roteiro de Aula

- Ponteiros
 - Declarando e utilizando ponteiros;
 - Ponteiro genérico;
 - Operações com ponteiros;
 - Ponteiros e arrays;
 - Ponteiro para ponteiro.

Ponteiros

- Ponteiros são variáveis “especiais” que não armazenam valores, mas endereços de memória;
- Os ponteiros tem como finalidade apontar um endereço de memória, facilitando assim a manipulação do dado armazenado no endereço.

Declarando Ponteiros

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

```
main(){  
    int numero = 30;  
    int *p_numero = NULL;  
  
    printf("%d", numero);  
    printf("%d", p_numero);  
}
```

Atribuindo valor para Ponteiros

```
#include <stdio.h>

main(){
    int numero = 30;
    int *pnumero = NULL;

    pnumero = &numero;

    printf("%d", numero); // 30
    printf("\n%d", &numero); // 6487620
    printf("\n%d", pnumero); // 6487620
    printf("\n%d", *pnumero); // 30
}
```

Alterando valor apontado

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

```
main(){  
    int numero = 30;  
    int *pnumero = NULL;  
  
    pnumero = &numero;  
  
    printf("%d", numero); // 30  
    printf("\n%d", &numero); // 6487620  
    printf("\n%d", pnumero); // 6487620  
    printf("\n%d", *pnumero); // 30  
  
    *pnumero = 12;  
    printf("\n%d", numero); // 12  
    printf("\n%d", *pnumero); // 12  
}
```

Operações com Ponteiros

- Um ponteiro só pode receber o endereço de uma variável do mesmo tipo do ponteiro;
- Sobre o valor de endereço armazenado por um ponteiro podemos apenas somar e subtrair valores inteiros.

int: 4 bytes

char: 1 byte

float: 4 bytes

double: 8 bytes

```
main(){  
    int numero = 12;  
    int *p = &numero;  
  
    printf("%d", p); // 6487620  
    p++;  
    printf("\n%d", p); // 6487624  
    p = p - 2;  
    printf("\n%d", p); // 6487616  
}
```

Ponteiro Genérico

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

```
main(){  
    int i = 10;  
    int *p;  
    void *pg;  
  
    p = &i;  
    pg = &i; // Endereco de int  
    printf("Endereco de i: %d \n", &i); // 6487620  
    printf("Endereco em p: %d \n", p); // 6487620  
    printf("Endereco em pg: %d \n", pg); // 6487620  
  
    pg = &p;  
    printf("Endereco de p: %d \n", &p); // 6487608  
    printf("Endereco em pg: %d \n", pg); // 6487608  
  
    pg = p;  
    printf("Endereco em p: %d \n", p); // 6487620  
    printf("Endereco em pg: %d \n", pg); // 6487620  
}
```


Ponteiro Genérico

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


```
main(){  
    int x = 10;  
    float y = 20;  
    void *p = NULL;  
  
    p = &x;  
    printf("Endereco de x: %d \n", &x); // 6487620  
    printf("Endereco em p: %d \n", p); // 6487620  
  
    p = &y;  
    printf("Endereco de y: %d \n", &y); // 6487616  
    printf("Endereco em p: %d \n", p); // 6487616  
}
```

Ponteiro Genérico: Exibindo valor apontado

```
#include <stdio.h>

main(){
    int x = 30;
    void *p = NULL;
    p = &x;
    printf("%d", *(int*)p);
}
```

Cast



Ponteiro Genérico: Operações

```
#include <stdio.h>

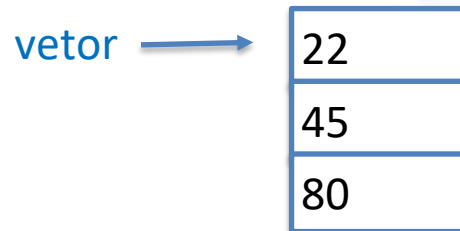
main(){
    int x = 10;
    void *p = &x;

    printf("p = %d \n", p); // 6487620
    p++;
    printf("p = %d \n", p); // 6487621
    p++;
    printf("p = %d \n", p); // 6487622
    p++;
    printf("p = %d \n", p); // 6487623
    p++;
}
```

Ponteiros e Arrays

- Arrays: Conjunto de dados armazenados de forma sequencial
 - Uma variável vetor é um ponteiro para onde ele começa.

```
int vetor[3] = {22, 45, 80};
```



Ponteiros e Arrays

```
#include <stdio.h>

main(){
    int vetor[3] = {22, 45, 80};
    int *p = vetor;

    printf("%d \n", vetor); // 6487600
    printf("%d \n", &vetor[0]); // 6487600
    printf("%d \n", p); // 6487600
}
```

Ponteiros e Arrays

```
#include <stdio.h>

main(){
    int vetor[3] = {22, 45, 80};
    int *p = vetor;

    printf("%d \n", vetor[0]); // 22
    printf("%d \n", *p); // 22
}
```

Ponteiros e Arrays

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

```
main(){  
    int vetor[3] = {22, 45, 80};  
    int *p = vetor;  
  
    printf("%d \n", vetor[1]); // 45  
    printf("%d \n", *(p+1)); // 45  
    printf("%d \n", p[1]); // 45  
}
```

Ponteiros e Arrays

```
#include <stdio.h>

main(){

    int *p = NULL, i;
    int array[5] = {1, 2, 3, 4, 5};

    p = array;

    for(i = 0; i < 5; i++){
        printf("%d", array[i]);
        printf("%d", p[i]);
        printf("%d", *(p+i));
    }

}
```


Array de Ponteiros

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

```
main(){
```

```
    int *p[2];
```

```
    int x = 12;
```

```
    int v[2] = {30, 10};
```

```
    p[0] = &x;
```

```
    p[1] = v;
```

```
    printf("%d \n", *p[0]); // 12
```

```
    printf("%d \n", p[1][1]); // 10
```

```
}
```

Ponteiro para Ponteiro

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

```
main(){
```

```
    int x = 12;
```

```
    int *p = &x;
```

```
    int **pp = &p;
```

```
    printf("\n%d", pp);
```

```
    printf("\n%d", *pp);
```

```
    printf("\n%d", **pp);
```

```
}
```



Faci facid FACIMP F&V fmf Presidência
Martha Falcão ISL UNIFAVIP UNI
METROCAMP RUY
BARBOSA | AREA1 UniF&V UniFanor