



# Linguagem de Programação

Passagem de Parâmetro – Referência

Gil Eduardo de Andrade





# Passagem de Parâmetro

## Introdução

- Na aula anterior foram abordados os conceitos iniciais sobre funções em C: declaração, chamada, implementação, retorno e parâmetros;
- A passagem de parâmetro abordada foi a “*por valor*”, onde o valor de uma variável é passado para função e copiado para dentro do parâmetro especificado em sua declaração;





# Passagem de Parâmetro

## Passagem “*por valor*”

- Quando utilizamos a passagem de parâmetro “*por valor*” a variável que é passada a função não tem seu valor modificado quando o código especificado para mesma é executado ;
- Observe o exemplo a seguir:



# Passagem de Parâmetro

## Exemplo de implementação (passagem “por valor”)

```
int somar(int a, int b);

int main() {
    int result, valor_a, valor_b;

    printf("Entre com valor a: ");
    scanf("%i", &valor_a);
    printf("Entre com valor b: ");
    scanf("%i", &valor_b);

    result = somar(valor_a, valor_b);
    printf("result = %i / valor_a = %i", result, valor_a);

    printf("\n");
    return 0;
}

int somar(int a, int b) {
    a = a + b;
    return a;
}
```

As variáveis “*valor\_a*” e “*valor\_b*” são passadas como parâmetro “por valor” para função *somar()*, onde o conteúdo armazenado por elas é copiado, respectivamente, para as variáveis (parâmetros) “*a*” e “*b*” da função *somar()*;

A variável (parâmetro) “*a*” tem seu valor alterado quando a função *somar()* é executada, porém o valor da variável “*valor\_a*” dentro da função *main()* não é alterado (vide próximo slide).

# Passagem de Parâmetro

## Exemplo de implementação (passagem “por valor”)

```
g1l3du4rd0@asus-ultrabook-g1l:/run/media/g1l3du4rd0/GEA/VirtualBox/Compartilhada/IFPR/2015/PC/au... x
Arquivo Editar Ver Pesquisar Terminal Ajuda
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc14]$ gcc por_valor.c
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc14]$ ./a.out
Entre com valor a: 2
Entre com valor b: 4
result = 6 / valor_a = 2
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc14]$
```

Observe que o conteúdo da variável “*valor\_a*” não é alterado, mantendo-se com valor igual a “2”, mesmo após a função *somar()* ter sido invocada e executada.



# Passagem de Parâmetro

## Passagem “*por referência*”

- A passagem de parâmetro “*por referência*” permite que, ao invés do valor, o endereço de memória (*referência*) de uma variável seja passado como parâmetro a uma função;
- Para que seja possível então receber esse endereço de memória (*referência*) o parâmetro da função deve ser um ponteiro;





# Passagem de Parâmetro

## Passagem “*por referência*”

- Sendo assim quando efetuamos a passagem de parâmetro “*por referência*” o ponteiro que recebe o endereço de memória da variável passada pode alterar o conteúdo da mesma, alterando seu valor ao final da execução da função;
- Observe o exemplo a seguir:



# Passagem de Parâmetro

## Exemplo de implementação (passagem “por referência”)

```
#include <stdio.h>

int somar(int *a, int b);

int main() {

    int result, valor_a, valor_b;

    printf("Entre com valor a: ");
    scanf("%i", &valor_a);
    printf("Entre com valor b: ");
    scanf("%i", &valor_b);

    result = somar(&valor_a, valor_b);
    printf("result = %i / valor_a = %i", result, valor_a);

    printf("\n");
    return 0;
}

int somar(int *a, int b) {
    *a = *a + b;
    return *a;
}
```

As variável “*valor\_a*” é passada como parâmetro “*por referência*” – observe a utilização do operador “&” – para função *somar()*, e seu endereço de memória passa a ser apontado pelo ponteiro “*a*”, parâmetro da função *somar()*;

O ponteiro (parâmetro) “*a*” altera o conteúdo – observe a utilização do operador “\*” – armazenado no endereço de memória apontado por ele (da variável *valor\_a*), logo o valor da variável “*valor\_a*” dentro da função *main()* é alterado (vide próximo slide).



# Passagem de Parâmetro

## Exemplo de implementação (passagem “por valor”)

```
g1l3du4rd0@asus-ultrabook-g1l:/run/media/g1l3du4rd0/GEA/VirtualBox/Compartilhada/IFPR/2015/PC/au... x
Arquivo Editar Ver Pesquisar Terminal Ajuda
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc14]$ ./a.out
Entre com valor a: 2
Entre com valor b: 4
result = 6 / valor_a = 6
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc14]$
```

Observe que o conteúdo da variável “*valor\_a*” é alterado, modificando seu valor de “2” para “6”, logo após a função *somar()* ter sido invocada e executada.



# Passagem de Parâmetro

## Vetores como parâmetro:

- Considerando o uso de variáveis ponteiro como parâmetros, torna-se possível também passar um conjunto de valores as funções;
- Esses conjuntos de valores podem ser, por exemplo, vetores (ponteiros alocados) com valores inteiros ou caracteres;





# Passagem de Parâmetro

## Vetores como parâmetro:

- Porém, ao invés de utilizar o operador “&” como fazemos na passagem “*por referência*” de uma variável, agora passamos um ponteiro já alocado e contendo um tamanho pré-definido;
- Observe os exemplos a seguir:



# Passagem de Parâmetro

## Exemplo de implementação (passagem “vetor char”)

```
void imprime_string(char *str);

int main() {

    char *nome = malloc(50*sizeof(char));

    printf("Digite um nome: ");
    gets(nome);
    // Passa a string nome (char *) para função
    imprime_string(nome); ←

    printf("\n");
    return 0;
}

void imprime_string(char *str) {
    // Imprime a string recebida
    printf("%s", str); ←
}
```

O ponteiro “*nome*” é passado para função *imprime\_string()*, logo o parâmetro “*str*”, que também é um ponteiro, passa a apontar para o endereço de memória da variável “*nome*”;

A função apresentada, via *printf()*, o conteúdo do endereço apontado por “*str*”, ou seja o nome digitado pelo usuário e armazenado no ponteiro “*nome*” (vide próximo slide). Perceba que, nesse caso, não é preciso utilizar o operador “*\**” para acessar o conteúdo desse endereço apontado por “*str*”;

# Passagem de Parâmetro

## Exemplo de implementação (passagem “vetor char”)

```
g1l3du4rd0@asus-ultrabook-g1l:/run/media/g1l3du4rd0/GEA/VirtualBox/Compartilhada/IFPR/20... x
Arquivo Editar Ver Pesquisar Terminal Ajuda
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc14]$ ./a.out
Digite um nome: Gil Eduardo
Gil Eduardo
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc14]$ ./a.out
Digite um nome: Ayrton Senna
Ayrton Senna
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc14]$ █
```

# Passagem de Parâmetro

## Exemplo de implementação (passagem “vetor char”)

```
int main() {  
  
    char *nome = malloc(50*sizeof(char));  
  
    printf("Digite um nome: ");  
    gets(nome);  
    // Passa a string nome (char *) para função  
    imprime_string(nome); ←  
    printf("nome = %s", nome);  
  
    printf("\n");  
    return 0;  
}  
  
void imprime_string(char *str) {  
    // Substitui posições pares por '@'  
    int a;  
    for(a=0; str[a]!='\0';a++) {  
        if(a%2==0)  
            str[a] = '@';  
    }  
    printf("str = %s\n", str);  
}
```

O ponteiro “*nome*” é passado para função *imprime\_string()*, logo o parâmetro “*str*”, que é um ponteiro, passa a apontar para o endereço de memória da variável “*nome*”;

A função *imprime\_string()* altera o conteúdo apontado pelo parâmetro “*str*” antes de apresentá-lo. Sendo assim o conteúdo da variável “*nome*”, local a função *main()*, também será alterado, como visto no conceito de passagem “*por referência*” (vide próximo slide).

# Passagem de Parâmetro

## Exemplo de implementação (passagem “vetor char”)

```
g1l3du4rd0@asus-ultrabook-g1l:/run/media/g1l3du4rd0/GEA/VirtualBox/Compartilhada/IFPR/20... x
Arquivo Editar Ver Pesquisar Terminal Ajuda
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc14]$ ./a.out
Digite um nome: Gil Eduardo
str = @i@ @d@a@d@
nome = @i@ @d@a@d@
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc14]$ ./a.out
Digite um nome: Ayrton Senna
str = @y@t@n@S@n@a
nome = @y@t@n@S@n@a
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc14]$
```

Observe que o conteúdo da variável “*nome*” também é alterado, modificando todas as suas posições pares o caractere ‘@’.

# Passagem de Parâmetro

## Exemplo de implementação (passagem “vetor int”)

```
void imprime_sena(int *sena);

int main() {

    int *s = malloc(6*sizeof(int));

    printf("Digite 6 números: ");
    scanf("%i %i %i %i %i %i", &s[0], &s[1], &s[2], &s[3], &s[4], &s[5]);
    // Passa o vetor 's' (int *) para função
    imprime_sena(s);

    printf("\n");
    return 0;
}

void imprime_sena(int *sena) {
    // Imprime o vetor recebido
    int a;
    for(a=0; a<6; a++) {
        printf("%i ", sena[a]);
    }
}
```

O ponteiro “s” é passado para função *imprime\_sena()*, logo o parâmetro “sena”, que também é um ponteiro, passa a apontar para o endereço de memória da variável “s”;

A função apresentada, via *printf()*, o conteúdo do endereço apontado por “sena”, ou seja o nome digitado pelo usuário e armazenado no ponteiro “nome” (vide próximo slide).





# Passagem de Parâmetro

## Exemplo de implementação (passagem “vetor int”)

```
g1l3du4rd0@asus-ultrabook-g1l:/run/media/g1l3du4rd0/GEA/VirtualBox/Compartilhada/IFPR/20... x
Arquivo Editar Ver Pesquisar Terminal Ajuda
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc14]$ ./a.out
Digite 6 números: 1 2 3 4 5 6
1 2 3 4 5 6
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc14]$ ./a.out
Digite 6 números: 9 8 7 6 5 4
9 8 7 6 5 4
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc14]$ █
```



# Passagem de Parâmetro

## Exemplo de implementação (passagem “vetor int”)

```
int main() {  
  
    int a;  
    int *s = malloc(6*sizeof(int));  
  
    printf("Digite 6 números: ");  
    scanf("%i %i %i %i %i %i", &s[0], &s[1], &s[2], &s[3], &s[4], &s[5]);  
    // Passa o vetor 's' (int *) para função  
    imprime_sena(s);  
    printf("s = ");  
    for(a=0; a<6; a++)  
        printf("%i ", s[a]);  
  
    printf("\n");  
    return 0;  
}  
  
void imprime_sena(int *sena) {  
    // Imprime o vetor recebido  
    int a;  
    printf("sena = ");  
    for(a=0; a<6; a++) {  
        sena[a] = sena[a] * 10;  
        printf("%i ", sena[a]);  
    }  
    printf("\n");  
}
```

O ponteiro “s” é passado para função *imprime\_sena()*, logo o parâmetro “sena”, que é um ponteiro, passa a apontar para o endereço de memória da variável “s”;

A função *imprime\_sena()* altera o conteúdo apontado pelo parâmetro “sena” antes de apresentá-lo. Sendo assim o conteúdo da variável “s”, local a função main(), também será alterado, como visto no conceito de passagem “por referência” (vide próximo slide).

# Passagem de Parâmetro

## Exemplo de implementação (passagem “vetor int”)

```
g1l3du4rd0@asus-ultrabook-g1l:/run/media/g1l3du4rd0/GEA/VirtualBox/Compartilhada/IFPR/20... x
Arquivo Editar Ver Pesquisar Terminal Ajuda
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc14]$ ./a.out
Digite 6 números: 1 2 3 4 5 6
sena = 10 20 30 40 50 60
s = 10 20 30 40 50 60
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc14]$ ./a.out
Digite 6 números: 9 8 7 6 5 4
sena = 90 80 70 60 50 40
s = 90 80 70 60 50 40
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc14]$
```

Observe que o conteúdo da variável “s” também é alterado, multiplicando todos os seus valores por **10**.



# Operadores de Condição

## Exemplos Utilizados no Documento

[http://www.gileduardo.com.br/ifpr/pcii/downloads/pc\\_exdoc14.zip](http://www.gileduardo.com.br/ifpr/pcii/downloads/pc_exdoc14.zip)

## Mais Exemplos sobre o Conteúdo

[http://www.gileduardo.com.br/ifpr/pcii/downloads/pc\\_ex14.zip](http://www.gileduardo.com.br/ifpr/pcii/downloads/pc_ex14.zip)

## Exercícios sobre o Conteúdo

[http://www.gileduardo.com.br/ifpr/lp/downloads/lp\\_pratica14.pdf](http://www.gileduardo.com.br/ifpr/lp/downloads/lp_pratica14.pdf)

