

Funções em C

SSC0300 - Linguagem de programação e aplicações

Prof. Dr Jó Ueyama

Estagiário PAE: Heitor Freitas

Instituto de Ciências Matemáticas e de Computação

Universidade de São Paulo

São Carlos - Brasil

01 de Setembro de 2014



Sumário

- 1 A linguagem C
- 2 Dúvidas

Exemplo de código em C

```
1  #include <stdio.h>
2
3  #define PI 3.1416
4
5  void printOla(void){
6      printf("Ola mundo!\n");
7  }
8
9  void printPI(void);
10
11 int main(void){
12     printOla();
13     printPI();
14     return 0;
15 }
16
17 void printPI(void){
18     printf("O valor de PI eh:%f\n", PI);
19 }
```

Destrinchando o exemplo em C

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

Inclusão de bibliotecas para estender as funcionalidades do programa. Neste caso, para o uso da *stdio.h*¹ para inclusão do *printf*².

```
#define PI 3.1416
```

Definição de constantes e macros.

```
void printOla(void){  
    printf("Ola mundo!\n");  
}  
  
void printPI(void);
```

Implementação e definição de funções.

¹<http://www.cplusplus.com/reference/cstdio>

²<http://www.cplusplus.com/reference/cstdio/printf>

Destrinchando o exemplo em C

A função *main* sempre deve retornar um número inteiro. Quando o programa executa sem erros, o retorno deve ser zero.

```
int main(void){  
    printOla();  
    printPI();  
    return 0;  
}
```

Implementação de função.

```
void printPI(void){  
    printf("O valor de PI eh:%f\n", PI);  
}
```

Função em C

Função em C é definida da seguinte maneira:

```
retorno nome_da_funcao(parametro)
{
    corpo_da_funcao;
}
```

O **retorno** pode ser de um dos seguintes tipos:

Tipos inteiros

Tipo	Faixa de valores
char	-128 a 127
unsigned char	0 a 255
signed char	-128 a 127
int	-2,147,483,648 a 2,147,483,647
unsigned int	0 a 4,294,967,295
short	-32,768 a 32,767
unsigned short	0 a 65,535
long	-2^{63} a 2^{63}
unsigned long	0 a 2^{63}

Alguns tipos de dados em C

Tipos de ponto flutuante:

Tipo	Faixa de valores
<code>float</code>	1.2^{-38} a 3.4^{+38}
<code>double</code>	2.3^{-308} a 1.7^{+308}
<code>long double</code>	3.4^{-4932} a 1.1^{+4932}

Exemplo de função

```
1 #include <stdio.h>
2 int main(int argc, char** argv){
3     int i;
4     printf("Quantidade de parametros: %d\n", argc);
5     for(i=0; i<argc; i++){
6         printf("Parametro %d:%s\n", i, argv[i]);
7     }
8     return 0;
9 }
```

A função `main` retorna um dado do tipo `int` (inteiro) com valor zero (`return 0;`). Os **parâmetros** dessa função³ são:

- `argc`: Variável do tipo inteiro que indica a quantidade de parâmetros passados por linha de comando;
- `argv`: Ponteiro para vetor de caracteres com os nomes dos parâmetros.

³<http://www.ime.usp.br/~pf/algoritmos/aulas/io.html#argc=argv>

O tipo `void`

O tipo `void` especifica que não há valor a ser retornado. Ele é utilizado em três tipos de situação:

- 1 **Retorno** de função: Quando não há dado a ser retornado;
- 2 Argumento de função: Quando não é aceito nenhum **parâmetro**.
- 3 Ponteiro para `void`: Um ponteiro do tipo `void *` representa o endereço de um objeto, mas não um tipo. Por exemplo a função de alocação dinâmica de memória **`void *malloc(size_t size)`**; retorna um ponteiro para `void`, o qual pode ser moldado para qualquer tipo de dado.

Exemplo de função

Outro exemplo:

```
1 void foo(void){  
2     printf("Ola mundo!\n");  
3 }
```

A função `foo` não retorna dado algum e não aceita argumentos.

Referências I

cplusplus.com (2014). <http://www.cplusplus.com/reference/cstdio/printf>,
acessado em 22 de Agosto de 2014.

cplusplus.com (2014). <http://www.cplusplus.com/reference/cstdio/>, acessado
em 22 de Agosto de 2014.

Dúvidas e sugestões

תודה
Dankie Gracias
Спасибо
Merci Takk
كشكراً
Köszönjök Terima kasih
Grazie Dziękujemy Dékojame
Ďakujeme Vielen Dank Paldies
Kiitos Täname teid 谢谢
Thank You Tak
感谢您 Obrigado Teşekkür Ederiz
Σας Ευχαριστούμ 감사합니다
ขอบคุณ
Bedankt Ďěkujeme vám
ありがとうございます
Tack