

# Programas Multiarquivos

Adriano Cruz  
adriano@nce.ufrj.br

Instituto de Matemática  
Departamento de Ciência da Computação  
UFRJ

28 de fevereiro de 2016

## Multiarquivos

O objetivo deste capítulo é apresentar uma introdução à criação de programas com vários arquivos.

# Section Summary

1 Duração

2 Escopo

3 Variáveis Globais

4 Módulos

5 Arquivos

- **Duração fixa** retém seus valores mesmo após a saída de seu escopo.
- **Duração automática** deixam de existir ao final da execução de seu escopo.
- Usa-se um especificador de classe de armazenamento para especificar a classe.
  - `auto` é usado para variáveis de duração automática. Como padrão variáveis locais a um bloco têm duração automática, raramente se usa a palavra **auto**.
  - `static` é usado para informar ao compilador que uma variável local deve ser tratada como de duração fixa.

# Duração Fixa

- Duração fixa cujo período de vida é todo o tempo de execução do programa.
- Duração fixa tem memória alocada para si no início do programa e permanece associada a esta mesma posição até o fim do programa.
- Um uso comum de variáveis de duração fixa é registrar o número de vezes que uma variável é executada.
- O escopo da variável continua o mesmo, não é alterado pelo fato da variável ser estática.

# Exemplo de variável automática

```
void quantasVezesFuiChamada(void)
{
    static unsigned contador = 0;
    ++contador;
    printf("Esta funcao foi chamada %u vez%s\n",
           contador,
           contador == 1 ? "." : "es.");
}
```

- A função imprime quantas vezes ela foi chamada.
- Em algumas linguagens não é permitido que uma variável local seja estática, neste caso a variável tem de ser global.

# Exemplo de variável automática

```
void Incrementa(void)
{
    int i = 1;
    static int j = 1;
    i++; j++;
    printf("Valor de i = %d, valor de j = %d\n", i, j);
}
```

Valor de i = 2, valor de j = 2

Valor de i = 2, valor de j = 3

Valor de i = 2, valor de j = 4

- Uma variável de duração fixa é iniciada apenas uma vez.
- Uma variável de duração automática é iniciada sempre que seu escopo é executado.

# Lembrar!

<b>Estática</b>	<b>Automática</b>
Iniciada implicitamente com zero	Não tem inicialização implícita
Iniciação não pode envolver variáveis	Iniciação pode envolver variáveis
Iniciada uma única vez	Pode ser iniciada várias vezes



# Section Summary

1 Duração

2 Escopo

3 Variáveis Globais

4 Módulos

5 Arquivos

# Escopo de Programa

- Um identificador com escopo de programa é ativo em todos os arquivos e blocos que compõem o programa.
- Apenas identificadores de variáveis e funções podem ter este tipo de escopo.
- Variáveis com escopo de programa são conhecidos como variáveis globais.
- Qualquer função não precedida por **static** tem escopo de programa.

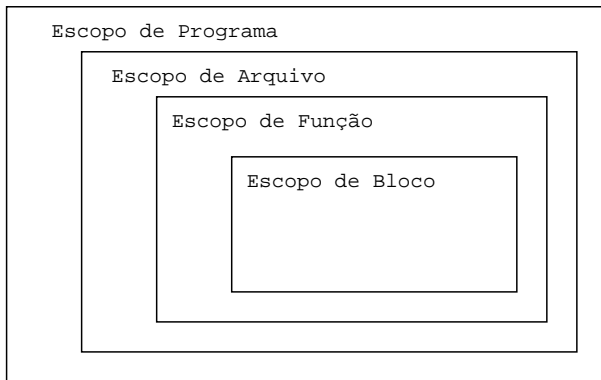
# Escopo de Arquivo

- Um identificador com escopo de arquivo tem validade em todos os blocos do arquivo no qual foi declarado a partir do ponto de declaração.
- Variáveis fora de funções e funções cujas definições sejam precedidas de **static** tem este tipo de escopo.
- **IMPORTANTE:** Aqui **static** não está ligado a duração da variável e sim ao escopo.
- Uma variável definida fora de função sem **static** é global.

# Escopo de Função

*Um identificador com escopo de função tem validade do início ao fim da função em que foi declarado.*

*Um identificador com escopo de bloco tem validade a partir do seu ponto de declaração até ao final do bloco no qual ele foi declarado.*



# Section Summary

- 1 Duração
- 2 Escopo
- 3 Variáveis Globais**
- 4 Módulos
- 5 Arquivos

# Recomendações

- **Variáveis globais devem ser evitadas.**
- Se for necessário procure usar uma convenção de nomes.
- Por exemplo use a letra g antes do nome (**double** gResultado).



**Definição:** a definição de uma variável global deve ser única em todo o programa e é responsável pela sua alocação em memória.

**Alusão:** Uma alusão é similar a uma definição e pode aparecer em vários arquivos que compõem o programa.

- Uma alusão não aloca memória para a variável.
- Uma alusão serve para informar ao compilador que a variável aludida é uma variável global definida em outro ponto do programa, talvez em outro arquivo.

- Sempre que for necessário utilizar uma variável global definida em um arquivo diferente daquele que em que se está trabalhando deve-se fazer uma alusão a variável.
- Para fazer uma alusão usa-se a palavra **extern**.
- Exemplo: **extern long int** gVarGlobal;

# Recomendações

- Para definir uma variável global, omita a palavra **extern** e inclua uma iniciação.
  - Exemplo: **long int** gVarGlobal = 0L;
- Para aludir a uma variável global, omita qualquer iniciação e inclua a palavra **extern**.
  - Exemplo: **extern long int** gVarGlobal;

# Funções Globais e Locais

- Uma função que somente é necessária no arquivo que contém a sua definição é chamada de função local ou estática.
- Usa-se **static** para funções locais.
- Uma função que é chamada em um arquivo diferente daquele que contém a sua definição é chamada função global ou externa.
- Usa-se **extern** para funções globais.
- Na ausência do especificador da classe da função (**extern** ou **static**) assume-se o especificador **extern**.

# Section Summary

1 Duração

2 Escopo

3 Variáveis Globais

**4 Módulos**

5 Arquivos

# Vantagens

- Programas ficam mais fáceis de entender, manter e depurar.
- O projeto pode ser dividido entre os componentes da equipe mais facilmente.
- Projeto mais fácil de ser gerenciado.

# Componentes do módulo

- Arquivos de cabeçalho (*header*) comumente tem a extensão **.h**
- Arquivos de programa (*source*) comumente tem a extensão **.c**

# Componentes do cabeçalho

- Alusões de funções
- Alusões de variáveis globais
- Declarações de tipos
- Definições de macros
- Definições de funções *inline*

*Cabeçalhos não geram código.*



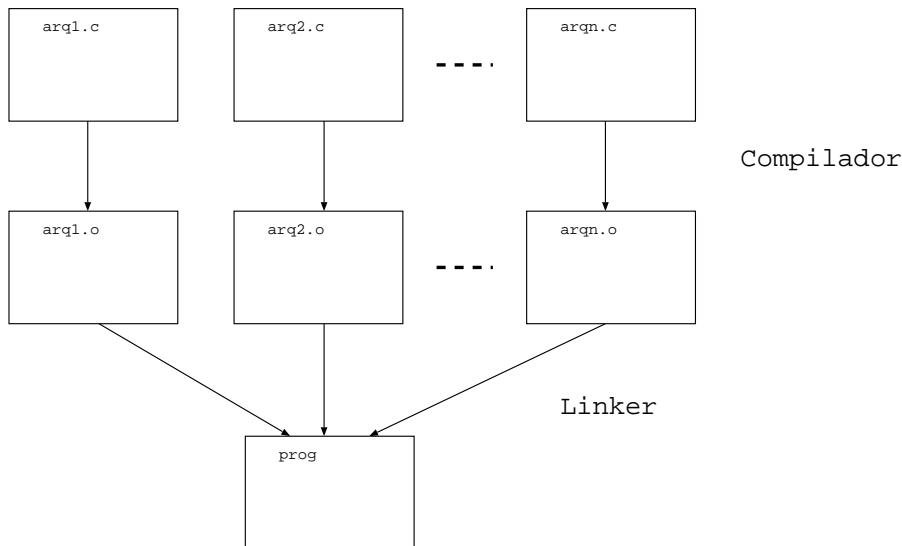
# Observações

- Para incluir um arquivo cabeçalho usar a diretiva **#include**
- **#include** <stdio.h>
- **#include** "Interface.h"
- Arquivos de programas não devem nunca ser incluídos em outro arquivo.

# Variáveis Globais

- Uma programa contendo muitas variáveis globais pode conter um arquivo próprio apenas para conter as variáveis globais.
- (`Globais.c`)
- Neste caso deve haver um arquivo contendo as alusões as variáveis globais.
- (`Globais.h`)

# Gerando Programa



Nome do Arquivo	Tipo do Modelo
main.c	Tipicamente contém apenas o main.c
Molde.h	Interface do módulo
Molde.c	Implementação do módulo
Globais.h	Alusões para variáveis globais
Globais.c	Definição e inicialização das variáveis globais
Defs.h	Definição de tipos e macros

# Section Summary

1 Duração

2 Escopo

3 Variáveis Globais

4 Módulos

5 Arquivos

```
/*
 * Titulo do programa
 * Autor do Crime
 * Data da Criação:
 * Ultima alteração
 * Descricao geral do programa
 *
 */
/* Inclua os arquivos de cabecalho necessarios */

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

```
/*
 * Interface do modulo Molde
 * Autor do Crime
 * Data da Criação:
 * Ultima alteração
 * Descricao do modulo
 */

#ifndef MOLDE_H
#define MOLDE_H

/* Incluir arquivos de cabecalhos necessarios */
/* Declaracoes de macros */
/* Declaracoes de tipos */
/* Alusoes de variaveis globais */
/* Alusoes de funcoes globais */

#endif
```

```
/*  
 * Implementacao do modulo Molde  
 * Autor do Crime  
 * Data da Criação:  
 * Ultima alteração  
 * Descricao do modulo  
 */
```

```
#include "Molde.h"
```

```
/* Incluir outros arquivos de cabecalhos necessarios */  
/* Definicoes de variaveis locais e globais do modulo */  
/* Definicoes de funcoes locais e globais do modulo */
```



```
/*
 * USAR ESTE SE FOR GRANDE O NUMERO DE GLOBAIS
 * Autor do Crime
 * Data da Criação:
 * Ultima alteração
 * Descricao das globais
 */
#ifndef GLOBAIS_H
#define GLOBAIS_H
/*
 * Incluir os arquivos de cabecalhos que contem as
 * definicoes dos tipos das variaveis globais
 * (SE FOR O CASO)
 *
 * Alusoes de variaveis globais comecando com
 * extern e sem iniciacoes
 */
#endif
```

```
/*  
 * USAR ESTE SE FOR GRANDE O NUMERO DE GLOBAIS  
 * Autor do Crime  
 * Data da Criação:  
 * Ultima alteração  
 * Descricao das variaveis globais  
 */
```

```
#include "Globais.h"
```

```
/*  
 * Definicoes das variaveis globais  
 * sem extern e com iniciacoes  
 */
```

```
/*
 * Autor do Crime
 * Data da Criação:
 * Ultima alteração
 * Descricao dos tipos e macros
 */
#ifndef DEFS_h
#define DEFS_H

/*
 * Inclua os cabecalhos necessarios
 *
 * Declaracoes de macros
 *
 * Declaracoes de tipos
 */

#endif
```

The End