



Introdução à Computação-I

Introdução à Linguagem C

Neste tópico abordaremos alguns conceitos básicos da linguagem C, como tipos de dados, comandos de entrada/saída e expressões aritméticas.

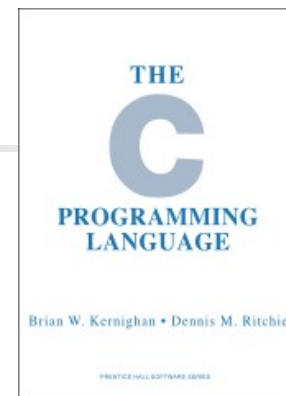
Prof. Ciro Cirne Trindade

Uma Visão Geral da Linguagem C

- Origens da linguagem C
 - BCLP → B por Ken Thompson
→ Linguagem C nos anos 70 por Dennis Ritchie
 - UNIX
 - Grande número de implementações
 - Compatibilidade



Uma Visão Geral da Linguagem C



- Versões do C
 - K&R C: em 1978, Brian Kernighan e Dennis Ritchie publicaram a 1ª edição do livro “The C Programming Language”
 - ANSI C (C89) e ISO C (C90): mesma linguagem
 - C99: introduziu várias mudanças
 - C11 (2011)
 - C18 (2018): última versão



Uma Visão Geral da Linguagem C

- Uma Linguagem de Médio Nível
 - Combina elementos das linguagens de nível alto com a funcionalidade da linguagem Assembly
 - Permite a manipulação de bits, bytes e endereços
 - C tem somente 44 palavras-chave, que são os comandos que formam a linguagem



Uma Visão Geral da Linguagem C

- Uma Linguagem Estruturada
 - Compartimentalização do código e dos dados
 - Sub-rotinas e blocos
 - Permite que sub-rotinas sejam declarados dentro de outras sub-rotinas
 - Possível em C a partir do padrão C99
 - Estruturas de controle
 - Comandos de decisão e repetição



Uma Visão Geral da Linguagem C

- Uma Linguagem p/ Programadores
 - C foi criada, influenciada e testada em campo por programadores reais
 - Poucas restrições, estruturas de blocos, funções independentes e um conjunto compacto de palavras-chave
 - Usada principalmente para a programação de software básico
 - Normalmente em lugar do Assembly



Uma Visão Geral da Linguagem C

- Compiladores vs. Interpretadores
 - Modo como um programa é executado
 - Interpretador:
 - lê o código fonte do programa uma linha por vez, executa as instruções específicas contidas naquela linha e depois busca a próxima linha
 - Compilador:
 - lê o programa inteiro e converte-o em código objeto (executável), que pode ser executado diretamente pelo computador



Uma Visão Geral da Linguagem C

- ... Compiladores vs. Interpretadores
 - A linguagem C é compilada
 - O compilador C mais conhecido e utilizado é o GNU gcc
 - Para compilar um arquivo fonte em C fazemos:
 - `gcc -o executável arquivo-fonte`
 - O código executável do C é conhecido por ser muito eficiente (execução rápida)



Uma Visão Geral da Linguagem C

- Os Fundamentos do Ambiente C
 - Todos os sistemas C são constituídos geralmente por três parte:
 - O ambiente de desenvolvimento
 - Editor de texto e compilador
 - IDE's como o CodeBlocks
 - A linguagem em si
 - A biblioteca padrão do C (C Standard Library)



Estrutura Básica de um Programa em C

- Um programa em C consiste em uma ou várias funções
- Forma geral das funções em C

```
tipo nome-da-função ([lista-de-parâmetros])
```

Tipo do
valor de
retorno
da
função

comandos

Informações
passadas à
função



Estrutura Básica de um Programa em C

- A Função `main()`
 - Todo programa em C deve possuir pelo menos uma função chamada `main`
 - Esta função é a primeira a ser executada, é o programa principal
 - Se um programa for constituído de uma única função esta será `main()`

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



Estrutura Básica de um Programa em C

- O primeiro programa em C

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

```
int main()
```

```
{
```

```
    printf("Ola, mundo!\n");
```

```
    return 0;
```

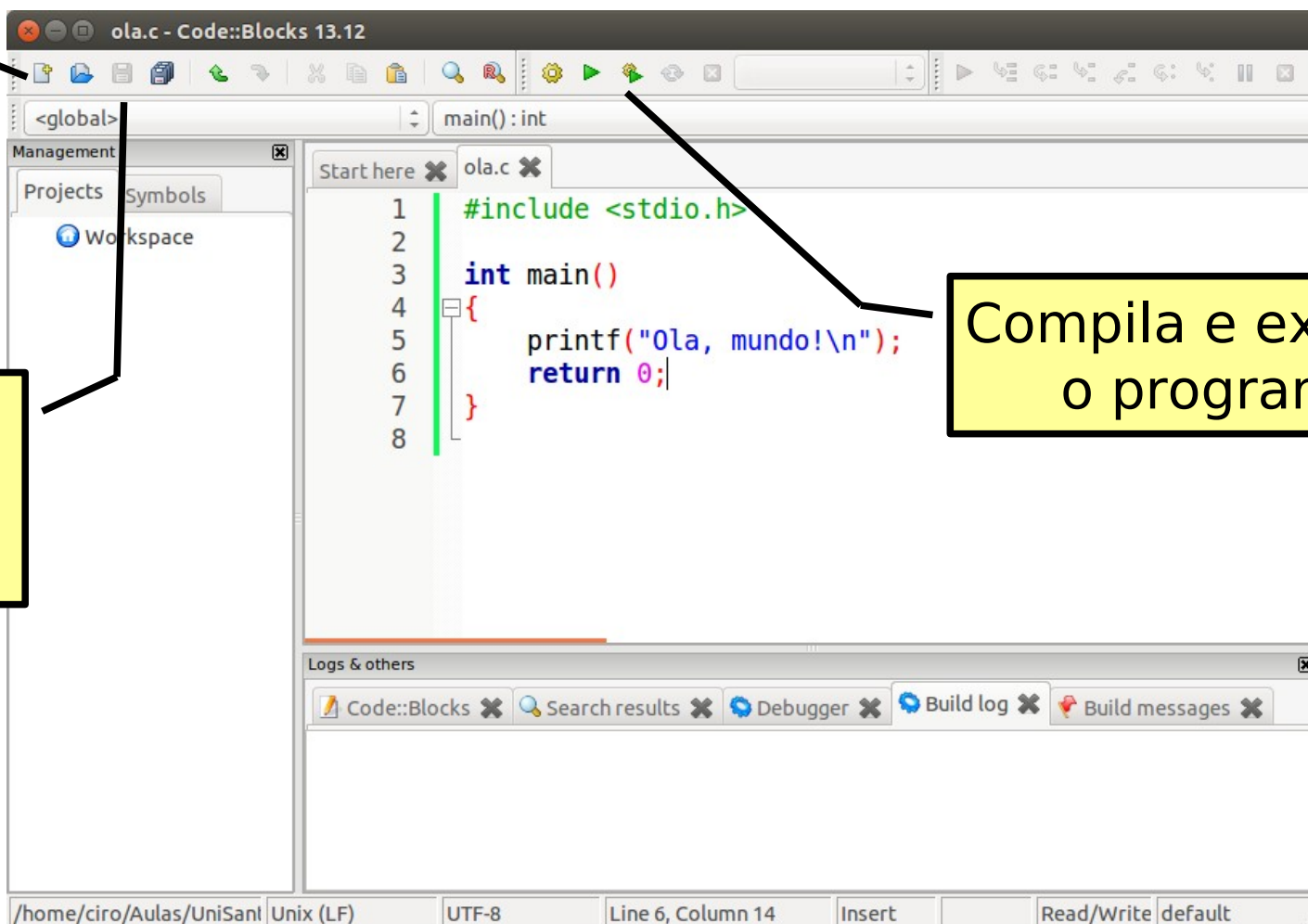
```
}
```



No Code::Blocks

Cria um
novo
arquivo
fonte

Salva o
arquivo
fonte (.c)



Compila e executa
o programa



Na linha de comando do Linux

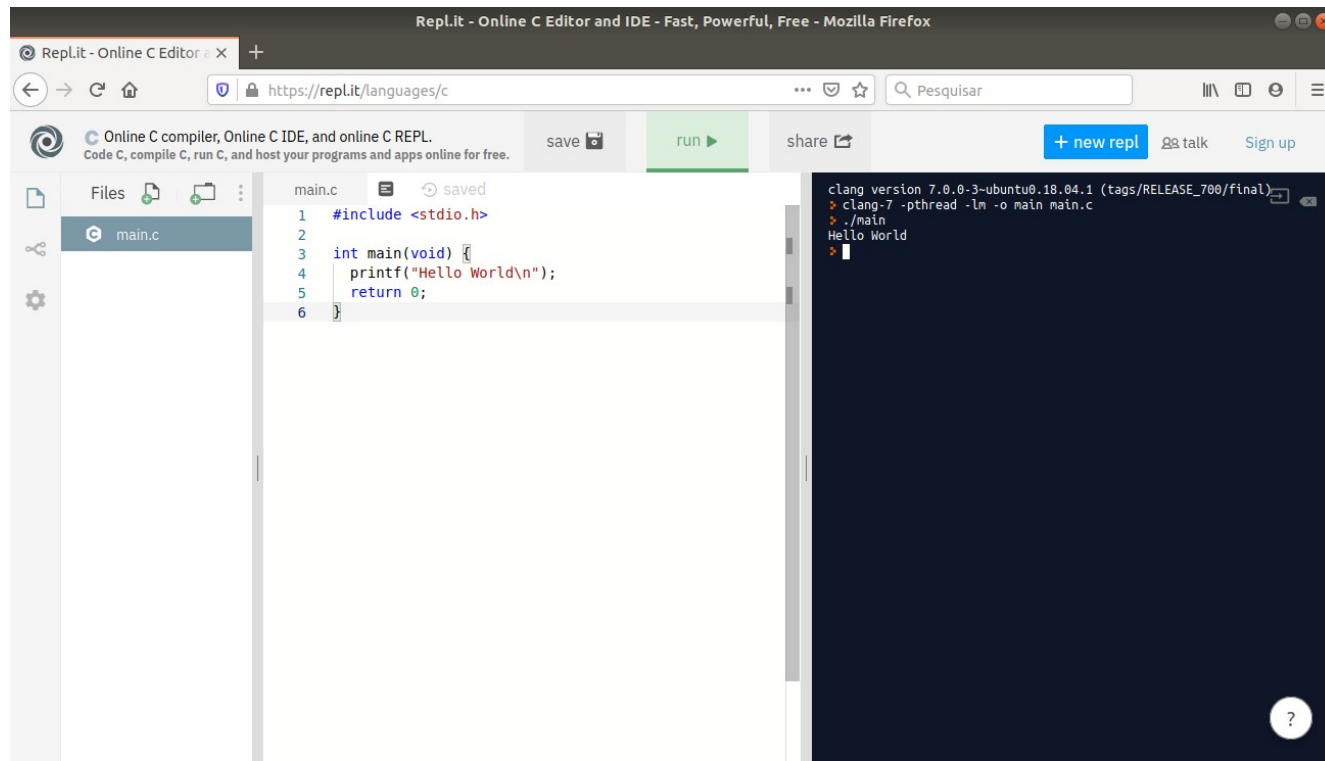
- Para compilar
 - `gcc -o ola ola.c`
- Para executar
 - `./ola`



-

Compiladores online

- Repl.it
 - <https://repl.it/languages/c>



The screenshot shows the Repl.it online C editor interface in a Mozilla Firefox browser. The browser tab is titled "Repl.it - Online C Editor and IDE - Fast, Powerful, Free - Mozilla Firefox". The address bar shows the URL "https://repl.it/languages/c". The page header includes the text "Online C compiler, Online C IDE, and online C REPL. Code C, compile C, run C, and host your programs and apps online for free." and buttons for "save", "run", "share", "+ new repl", "talk", and "Sign up". The left sidebar shows a file explorer with "main.c" selected. The main editor area displays the following C code:

```
1 #include <stdio.h>
2
3 int main(void) {
4     printf("Hello World\n");
5     return 0;
6 }
```

The right sidebar shows the output of the program, which is "Hello World". The output also includes the compiler version information: "clang version 7.0.0-3-ubuntu0.18.04.1 (tags/RELEASE_700/final)".

Tipos de Dados em C

- Existem 7 tipos de dados básicos em C

Tipo	Tamanho	Faixa
_Bool	8 bits	0 ou 1
char	8 bits	-128 a 127
int	32 bits	-2.147.483.648 a 2.147.483.647
long long	64 bits	-9.223.372.036.854.775.808 a 9.223.372.036.854.775.807
float	32 bits	$3,4 \times 10^{-38}$ a $3,4 \times 10^{38}$
double	64 bits	$1,7 \times 10^{-308}$ a $1,7 \times 10^{308}$
void	0	sem valor



Tipos de Dados em C

- Modificadores (**unsigned**, **short** e **long**)

Tipo	Tamanho	Faixa
unsigned char	8 bits	0 a 255
unsigned int	32 bits	0 a 4.294.967.295
short int	16 bits	-32.768 a 32.767
unsigned long int	32 bits	0 a 4.294.967.295
unsigned short int	16 bits	0 a 65.535
long int	32 bits	-2.147.483.648 a 2.147.483.647
unsigned long long	64 bits	0 a 18.446.744.073.709.551.615
long double	80 bits	$3,4 \times 10^{-4932}$ a $1,1 \times 10^{+4932}$

Nomes de Variáveis

- Devem começar sempre por uma letra ou o caractere de sublinhar (_), os demais podem ser letras, números ou caracteres de sublinhar
- Uma variável não pode ter o mesmo nome de uma palavra-chave em C
- Letras minúsculas e maiúsculas são diferentes:
 - Peso, PESO, peso, PeSo



Palavras-chaves em C

- auto
- break
- case
- char
- const
- continue
- default
- do
- double
- else
- enum
- extern
- float
- for
- goto
- if
- inline
- int
- long
- register
- restricted
- return
- short
- signed
- sizeof
- static
- struct
- switch
- typedef
- union
- unsigned
- void
- volatile
- while
- _Alignas
- _Alignof
- _Atomic
- _Bool
- _Complex
- _Generic
- _Imaginary
- _Noreturn
- _Static_assert
- _Thread_local



Exercício

- Quais dos nomes baixo são nomes válidos de variáveis em C abaixo?

() _kbytes

() 2nota

() hot-dog

() não

() taxa de juros

() “valido”

() int

() Float



Declarando variáveis

- Sintaxe:

```
tipo lista-de-variáveis;
```

- Exemplos:

```
int peso, idade;
```

```
float altura, salario;
```

```
char opcao;
```

Inicializando Variáveis

- É possível combinar uma declaração de variável com o operador de atribuição

```
/* inicialização de variáveis */  
#include <stdio.h>
```

Comentário em bloco

```
int main() {  
    int evento = 5; // número da competição  
    char corrida = 'C';  
    float tempo = 27.25;  
    ...  
    return 0;  
}
```

Comentário em linha

Constantes do tipo caractere aparecem entre apóstrofes

O separador da parte inteira parte fracionária em números reais é o ponto (.)



A Função `printf()`

- Uma das funções de saída usadas em C
- Não é um comando e sim uma função definida no arquivo de cabeçalho `stdio.h`
- Para usar a `printf()` em um programa em C é preciso incluir este arquivo
- Nós incluimos um arquivo de cabeçalho em um programa em C através da diretiva `#include`:
`#include <nome-do-arquivo>`



A Função `printf()`

- Entre os parênteses da função `printf()` são colocadas as informações que desejamos que sejam exibidas no vídeo
- A sintaxe da função `printf()` é a seguinte:

```
printf ("expressão-de-controle",  
        lista-de-argumentos) ;
```



A Função `printf()`

- Códigos de C para controle de fluxo do texto e caracteres especiais

Códigos Especiais	Significado
<code>\a</code>	Alerta. Soa o <i>beep</i> do sistema.
<code>\b</code>	Retrocesso. Retrocede um espaço.
<code>\n</code>	Nova linha. Posiciona o curso no início da nova linha.
<code>\r</code>	Retorno de cursor (CR). Posiciona o cursor no início da linha atual; não avança para a próxima linha.
<code>\t</code>	Tabulação. Move o cursor para a próxima de tabulação.
<code>\v</code>	Tabulação vertical.
<code>\"</code>	Aspas. Imprime um caractere de aspas duplas.
<code>\\</code>	Barra. Imprime um caractere de barra invertida.
<code>%%</code>	Imprime um único %

A Função `printf()`

- Códigos para impressão formatada

Código	Tipo de Dados	Formato
<code>%c</code>	<code>char</code> ou <code>int</code>	Caractere
<code>%d</code>	<code>int</code> ou <code>char</code>	Inteiro decimal
<code>%i</code>	<code>int</code> ou <code>char</code>	Inteiro
<code>%Ld</code> ou <code>%lld</code>	<code>long long</code>	Decimal
<code>%e</code>	<code>float</code> ou <code>double</code>	Notação científica
<code>%f</code>	<code>float</code>	Ponto flutuante
<code>%lf</code>	<code>double</code>	Ponto flutuante
<code>%o</code>	<code>int</code> ou <code>char</code>	Octal
<code>%s</code>	<code>char[]</code>	Cadeia de caracteres
<code>%x</code>	<code>int</code> ou <code>char</code>	Hexadecimal

A Função `printf()`

- Códigos para tipos com modificadores

Código	Tipo de Dados	Formato
<code>%Lf</code>	long double	Ponto flutuante
<code>%lu</code>	unsigned long int	Decimal
<code>%ld</code>	long int	Decimal
<code>%u</code>	unsigned int	Decimal
<code>%hd</code>	short int	Decimal



A Função `printf()`

- Tamanhos de campos na impressão
 - em `printf()` é possível estabelecer o tamanho mínimo para a impressão de um campo
- os campos podem ser alinhados à esquerda ou a direita

```
printf("%10.2f %10.2f %10.2f\n", 8.0, 15.3, 584.13);  
printf("%-10.2f %-10.2f %-10.2f\n", 8.0, 15.3,  
      584.13);
```



A Função `printf()`

- Complementando com zeros a esquerda

```
#include <stdio.h>
int main()
{
    printf("%04d\n", 21);
    printf("%06d\n", 21);
    printf("%06.2f\n", 21.0);
    return 0;
}
```



A Função `scanf()`

- Uma das funções de entrada da linguagem C
- Permite ler dados formatados do teclado
- Sintaxe:

```
scanf ("expressão-de-controle",  
        lista-de-argumentos) ;
```



A Função `scanf()`

- O Operador de Endereço (&)
 - toda variável ocupa uma certa localização na memória, e seu endereço é o do primeiro byte ocupado por ela
 - seja `n` uma variável, `&n` devolverá o endereço do primeiro byte onde `n` está guardada
 - por exemplo:
`scanf("%d", &n);`

A Função `scanf()`

- A função `scanf()` devolve o número de itens lidos

- Por exemplo:

```
int a, b;
```

```
printf("%d\n", scanf("%d %d", &a, &b));
```

Devolve 2 se o usuário
informar 2 números inteiros

- A função `scanf()` entende que o espaço em branco ou o Enter finalizam uma entrada de dados

Expressões Aritméticas

- Operadores aritméticos

Símbolo	Operação
+	Soma
-	Subtração
*	Multiplicação
/	Divisão
%	Módulo (resto da divisão inteira)



O operador de divisão: /

- Se os operandos forem inteiros, o resultado é o quociente inteiro da divisão
- Se pelo menos um dos operandos for `float` ou `double`, o resultado é um `float` ou `double`



O operador de divisão: /

```
#include <stdio.h>

int main() {
    printf("%d / %d = %d\n", 5, 2, 5 / 2);
    // 5 / 2 = 2
    printf("%d / %.1f = %.1f\n", 5, 2.,
           5 / 2.);
    // 5 / 2.0 = 2.5
    return 0;
}
```

Expressões Aritméticas

- Prioridade dos operadores aritméticos

Prioridade	Operadores
1º	*, / e %
2º	+ e -

- A ordem de prioridade dos operadores pode ser quebrada com o uso dos parênteses

Expressões Aritméticas

- Exemplo:

- Álgebra

- $m = \frac{a+b+c+d+e}{5}$

- C:

- $m = (a + b + c + d + e) / 5$

- Prioridade

- $y = a * x * x + b * x + c;$

1º 2º 4º 3º 5º



Exemplo: Programa para converter uma temperatura em graus Fahrenheit para graus Celsius

```
include <stdio.h>
int main()
{
    int ftemp, ctemp;
    printf("Digite a temperatura em graus
Fahrenheit: ");
    scanf("%d", &ftemp);
    ctemp = (ftemp - 32) * 5 / 9;
    printf("Temperatura em graus Celsius: %d\n",
        ctemp);
    return 0;
}
```



Funções da biblioteca matemática (`math.h`)

Função	Descrição	Exemplo
<code>sqrt(x)</code>	Raiz quadrada de x	<code>sqrt(900.0)</code> é 30.0
<code>exp(x)</code>	Função exponencial e^x	<code>exp(1.0)</code> é 2.718282
<code>log(x)</code>	Logaritmo natural de x	<code>log(2.718282)</code> é 1.0
<code>log10(x)</code>	Logaritmo na base 10 de x	<code>log10(1.0)</code> é 0.0
<code>ceil(x)</code>	Arredonda x para o menor inteiro maior que x	<code>ceil(9.2)</code> é 10.0 <code>ceil(9.8)</code> é 10.0
<code>floor(x)</code>	Arredonda x para o maior inteiro menor que x	<code>floor(9.2)</code> é 9.0 <code>floor(9.8)</code> é 9.0
<code>pow(x, y)</code>	x elevado à potência y (x^y)	<code>Pow(2, 5)</code> é 32.0



Funções da biblioteca matemática (`math.h`)

- Além de funções matemáticas, a `math.h` possui também a definição de algumas constantes, tais como π (`M_PI`) e o número de Euler e (`M_E`)



Exemplo: Programa para calcular a área de um círculo dado seu raio

```
#include <stdio.h>
#include <math.h>
int main()
{
    int raio;
    printf("Digite o raio do círculo: ");
    scanf("%d", &raio);
    printf("Área do círculo: %lf\n",
           M_PI * pow(raio, 2));
    return 0;
}
```

Referências

- SCHILDT, Herbert. *C Completo e Total*. 3. ed., Makron Books, 1997.
- DEITEL, H.M.; DEITEL, P.J.. *Como Programar em C*. 2. ed., LTC, 1999.
- MIZRAHI, V.V.. *Treinamento em Linguagem C*. Módulo 1, Makron Books, 1990.
- PRATA, Stephen. *C Primer Plus*. 6. ed., Addison Wesley, 2014.