

Introdução à Programação | IP-NBC

IP-NBC - Aula 02

Gustavo Teodoro Laureano

gustavo@inf.ufg.br

www.inf.ufg.br/~gustavo

sala 112

Universidade Federal de Goiás (UFG)

Instituto de Informática (INF)

www.inf.ufg.br

INSTITUTO DE
INFORMÁTICA
UFG

Agenda da apresentação

Contextualização

Raciocínio computacional

Definição de Algoritmo

Organização de um computador

Representação de dados

Linguagem de Programação

Estruturas Básicas de uma Linguagem de Programação

Contextualização

Por que estudar computação?

O mundo moderno é **dependente** da computação!

Isso é um **fato**.



Por que estudar computação?

O mundo moderno é **dependente** da computação!

Isso é um **fato**.

Exemplos de serviços dependentes da computação

- Sistemas de comunicação
- Internet
- Sistemas bancários
- Governo
- Redes sociais

Por que estudar computação?

O mundo moderno é **dependente** da computação!
Isso é um **fato**.

Exemplos de serviços dependentes da computação

- Sistemas de comunicação
- Internet
- Sistemas bancários
- Governo
- Redes sociais

Agora, imagine sua vida neste momento sem a existência de computadores!

Qual o papel dos sistemas computacionais?

- Automatização de processos
- Organização de informação
- Solução de problemas



Qual o papel dos sistemas computacionais?

- Automatização de processos
- Organização de informação
- Solução de problemas

No entanto, a computação vai **além** do nível das aplicações!

Qual o papel dos sistemas computacionais?

- Automatização de processos
- Organização de informação
- Solução de problemas

No entanto, a computação vai **além** do nível das aplicações!

Um **Cientista da Computação** tem a capacidade de promover o desenvolvimento da própria computação.

INSTITUTO DE
INFORMÁTICA
UFG

Computadores: Hardware vs Software

Computadores

Um computador é um dispositivo capaz de receber instruções que visam processar dados por meio de cálculos e decisões lógicas.

O equipamento físico, que permite a realização das instruções, é denominado **Hardware**.

Atualmente, esses equipamentos são predominantemente eletrônicos, mas também há versões mecânicas.

Software

Lista finita de instruções computacionais que controlam o funcionamento do Hardware.

Raciocínio computacional

Exemplos de problemas de lógica

- Temos 4 valores numéricos disponíveis. Descreva a sequência de passos para calcular a média desses valores.



Exemplos de problemas de lógica

- Temos 4 valores numéricos disponíveis. Descreva a sequência de passos para calcular a média desses valores.
- Temos 9 moedas de 1 real. Dessas moedas, uma é falsa. Sabemos que as moedas verdadeiras de mesmo valor possuem o mesmo peso. Supondo que temos uma balança de equilíbrio disponível, descreva a sequência de passos para identificarmos a moeda falsa.

INSTITUTO DE
INFORMÁTICA
UFG

Exemplos de problemas de lógica

- Temos 4 valores numéricos disponíveis. Descreva a sequência de passos para calcular a média desses valores.
- Temos 9 moedas de 1 real. Dessas moedas, uma é falsa. Sabemos que as moedas verdadeiras de mesmo valor possuem o mesmo peso. Supondo que temos uma balança de equilíbrio disponível, descreva a sequência de passos para identificarmos a moeda falsa.
- Supondo que a moeda falsa é mais pesada, resolva o problema anterior utilizando somente 2 pesagens. Descreva o algoritmo utilizado.

INSTITUTO DE
INFORMÁTICA
UFG

Definição de Algoritmo

Suponha que você deseje fazer um bolo simples. Você deverá seguir uma receita, ou uma sequência de passos para atingir seu objetivo.

Você precisará de:

- Ferramentas
- Ingredientes
- Procedimento

Receita de um bolo simples [2]:

Ingredientes

- 2 xícaras (chá) de açúcar
- 3 xícaras (chá) de farinha de trigo
- 4 colheres (sopa) de margarina
- 3 ovos
- 1 e 1/2 xícara (chá) de leite
- 1 colher (sopa) bem cheia de fermento em pó



Receita de um bolo simples [2]:

Ingredientes

- 2 xícaras (chá) de açúcar
- 3 xícaras (chá) de farinha de trigo
- 4 colheres (sopa) de margarina
- 3 ovos
- 1 e 1/2 xícara (chá) de leite
- 1 colher (sopa) bem cheia de fermento em pó

Ferramentas

- 1 xícara
- 1 colher de sopa
- 1 garfo
- 1 bacia
- 1 assadeira
- 1 forno

Receita de um bolo simples [2]:

Modo de preparo

1. Bata as claras em neve e reserve
2. Misture as gemas, a margarina e o açúcar até obter uma massa homogênea
3. Acrescente o leite e a farinha de trigo aos poucos sem parar de bater
4. Por último, adicione as claras em neve e o fermento
5. Despeje a massa em uma forma grande de furo central untada e enfarinhada
6. Asse em forno médio 180 °C, preaquecido por aproximadamente 40 minutos ou ao furar com um garfo, este saia limpo

Exemplo de Algoritmo

Suponha o problema de encontrar a raiz quadrada do número x ; Sendo $y = \sqrt{x}$, qual o valor de y ?



Exemplo de Algoritmo

Suponha o problema de encontrar a raiz quadrada do número x ; Sendo $y = \sqrt{x}$, qual o valor de y ?

Tipos de conhecimento

- Declarativo: Se $x = 121$, então $y = 11$ porque $11^2 = 121$. Isso funciona para os quadrados que você tem memorizado.
- Procedimental: Qual a sequência de passos matemáticos que nos permite calcular a raiz quadrada de qualquer número x ?

INSTITUTO DE
INFORMÁTICA
UFG

Exemplo de Algoritmo

Suponha o problema de encontrar a raiz quadrada do número x ; Sendo $y = \sqrt{x}$, qual o valor de y ?

Tipos de conhecimento

- Declarativo: Se $x = 121$, então $y = 11$ porque $11^2 = 121$. Isso funciona para os quadrados que você tem memorizado.
- Procedimental: Qual a sequência de passos matemáticos que nos permite calcular a raiz quadrada de qualquer número x ?

Por exemplo, o **Método Babilônico**: Inicie y_k com um número qualquer e execute

$$y_k = \frac{y_{k-1} + \frac{x}{y_{k-1}}}{2} \quad (1)$$

enquanto que $|y_k^2 - x| > e$, onde e é a precisão desejada para o cálculo.

Qual a definição de Algoritmo?

Algoritmo é uma **sequência finita** de **instruções** bem definidas e não ambíguas, cada uma das quais devendo ser executadas mecânica ou eletronicamente em um intervalo de tempo finito e com uma quantidade de esforço finita.

– Wikipedia [3]

Algoritmo é a descrição de um conjunto de comandos que, obedecidos, resultam numa sucessão finita de **ações**.

– Harry Farrer[1]

INSTITUTO DE
INFORMÁTICA
UFG

Qual a definição de Algoritmo?

Algoritmo é uma **sequência finita** de **instruções** bem definidas e não ambíguas, cada uma das quais devendo ser executadas mecânica ou eletronicamente em um intervalo de tempo finito e com uma quantidade de esforço finita.

– Wikipedia [3]

Algoritmo é a descrição de um conjunto de comandos que, obedecidos, resultam numa sucessão finita de **ações**.

– Harry Farrer[1]

O que é Instrução/Ação?

É um acontecimento que, a partir de um estado inicial, após um período de tempo finito, produz um estado final previsível e bem definido.

– Harry Farrer[1]

Organização de um computador

Elementos de um computador

- Unidade de entrada
- Unidade de saída
- Memória
- Barramento
- Central de processamento (CPU)
 - Unidade de Controle
 - Unidade Lógica e Aritimética (ULA)
 - Registradores

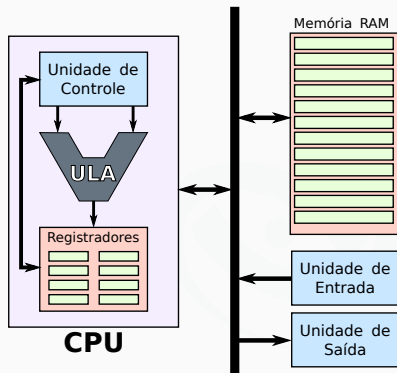


Figura 1: Organização de um Computador.

Representação de dados

Há basicamente dois tipos de informações processadas em um computador: informações numéricas/quantitativas e simbólicas.

- **Numéricas:** refere-se à representação de quantidades usando números inteiros e reais.
- **Simbólica** refere-se à informações que possuem uma representação simbólica, tais como dígitos e caracteres em geral.

Ambos os casos são tratados como números dentro de um computador.

Representação quantitativa: Decimal, Hexadecimal e Binária

No nosso dia a dia, estamos acostumados a usar o sistema **Decimal** de contagem para representar quantidades mas em um computador a informação é representada por 0 e 1 (**Binária**) e algumas vezes no formato **Hexadecimal**. Assim temos:

Decimal: usa 10 símbolos
{0,1,2,3,4,5,6,7,8,9}

Hexadecimal: usa 16 símbolos
{0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F}

Binário: usa 2 símbolos {0,1}

| Decimal | Hex | Binária |
|---------|-----|---------|
| 0 | 0 | 0 |
| 1 | 1 | 1 |
| 2 | 2 | 10 |
| 3 | 3 | 11 |
| 4 | 4 | 100 |
| 5 | 5 | 101 |
| 6 | 6 | 110 |
| 7 | 7 | 111 |
| 8 | 8 | 1000 |
| 9 | 9 | 1001 |
| 10 | A | 1010 |
| 11 | B | 1011 |
| 12 | C | 1100 |
| 13 | D | 1101 |
| 14 | E | 1110 |
| 15 | F | 1111 |

Representação de símbolos: Tabela ASCII

| Decimal | Hex | Char | Decimal | Hex | Char | Decimal | Hex | Char | Decimal | Hex | Char |
|---------|-----|------------------------|---------|-----|---------|---------|-----|------|---------|-----|-------|
| 0 | 0 | [NULL] | 32 | 20 | [SPACE] | 64 | 40 | @ | 96 | 60 | ` |
| 1 | 1 | [START OF HEADING] | 33 | 21 | ! | 65 | 41 | A | 97 | 61 | a |
| 2 | 2 | [START OF TEXT] | 34 | 22 | " | 66 | 42 | B | 98 | 62 | b |
| 3 | 3 | [END OF TEXT] | 35 | 23 | # | 67 | 43 | C | 99 | 63 | c |
| 4 | 4 | [END OF TRANSMISSION] | 36 | 24 | \$ | 68 | 44 | D | 100 | 64 | d |
| 5 | 5 | [ENQUIRY] | 37 | 25 | % | 69 | 45 | E | 101 | 65 | e |
| 6 | 6 | [ACKNOWLEDGE] | 38 | 26 | & | 70 | 46 | F | 102 | 66 | f |
| 7 | 7 | [BELL] | 39 | 27 | ' | 71 | 47 | G | 103 | 67 | g |
| 8 | 8 | [BACKSPACE] | 40 | 28 | (| 72 | 48 | H | 104 | 68 | h |
| 9 | 9 | [HORIZONTAL TAB] | 41 | 29 |) | 73 | 49 | I | 105 | 69 | i |
| 10 | A | [LINE FEED] | 42 | 2A | * | 74 | 4A | J | 106 | 6A | j |
| 11 | B | [VERTICAL TAB] | 43 | 2B | + | 75 | 4B | K | 107 | 6B | k |
| 12 | C | [FORM FEED] | 44 | 2C | , | 76 | 4C | L | 108 | 6C | l |
| 13 | D | [CARRIAGE RETURN] | 45 | 2D | - | 77 | 4D | M | 109 | 6D | m |
| 14 | E | [SHIFT OUT] | 46 | 2E | . | 78 | 4E | N | 110 | 6E | n |
| 15 | F | [SHIFT IN] | 47 | 2F | / | 79 | 4F | O | 111 | 6F | o |
| 16 | 10 | [DATA LINK ESCAPE] | 48 | 30 | 0 | 80 | 50 | P | 112 | 70 | p |
| 17 | 11 | [DEVICE CONTROL 1] | 49 | 31 | 1 | 81 | 51 | Q | 113 | 71 | q |
| 18 | 12 | [DEVICE CONTROL 2] | 50 | 32 | 2 | 82 | 52 | R | 114 | 72 | r |
| 19 | 13 | [DEVICE CONTROL 3] | 51 | 33 | 3 | 83 | 53 | S | 115 | 73 | s |
| 20 | 14 | [DEVICE CONTROL 4] | 52 | 34 | 4 | 84 | 54 | T | 116 | 74 | t |
| 21 | 15 | [NEGATIVE ACKNOWLEDGE] | 53 | 35 | 5 | 85 | 55 | U | 117 | 75 | u |
| 22 | 16 | [SYNCHRONOUS IDLE] | 54 | 36 | 6 | 86 | 56 | V | 118 | 76 | v |
| 23 | 17 | [ENG OF TRANS. BLOCK] | 55 | 37 | 7 | 87 | 57 | W | 119 | 77 | w |
| 24 | 18 | [CANCEL] | 56 | 38 | 8 | 88 | 58 | X | 120 | 78 | x |
| 25 | 19 | [END OF MEDIUM] | 57 | 39 | 9 | 89 | 59 | Y | 121 | 79 | y |
| 26 | 1A | [SUBSTITUTE] | 58 | 3A | : | 90 | 5A | Z | 122 | 7A | z |
| 27 | 1B | [ESCAPE] | 59 | 3B | ; | 91 | 5B | [| 123 | 7B | { |
| 28 | 1C | [FILE SEPARATOR] | 60 | 3C | < | 92 | 5C | \ | 124 | 7C | |
| 29 | 1D | [GROUP SEPARATOR] | 61 | 3D | = | 93 | 5D |] | 125 | 7D | } |
| 30 | 1E | [RECORD SEPARATOR] | 62 | 3E | > | 94 | 5E | ^ | 126 | 7E | ~ |
| 31 | 1F | [UNIT SEPARATOR] | 63 | 3F | ? | 95 | 5F | _ | 127 | 7F | [DEL] |

Figura 2: Tabela ASCII. Fonte [4].

Linguagem de Programação

Como programar um computador?

- Na prática, um computador **entende** somente a sua própria **linguagem de máquina** que é definida pela sua própria arquitetura de *hardware*.
- No entanto, programadores escrevem códigos em várias **linguagens de programação**, o que exige uma fase de tradução do código antes dele ser efetivamente executado por um computador.

INSTITUTO DE
INFORMÁTICA
UFG

Como programar um computador?

- Na prática, um computador **entende** somente a sua própria **linguagem de máquina** que é definida pela sua própria arquitetura de *hardware*.
- No entanto, programadores escrevem códigos em várias **linguagens de programação**, o que exige uma fase de tradução do código antes dele ser efetivamente executado por um computador.
- O **Compilador** é um programa de computador responsável por transformar um código escrito em uma **linguagem compilável** e transformá-lo em um **código semanticamente equivalente** em outra linguagem de programação.

- **Linguagem de máquina:** código binário, ilegível, que contém instruções escritas diretamente na codificação entendida pelo processador.
- **Linguagem Assembly:** código textual, legível, que contém instruções que são transcrições textuais dos códigos binários entendidos pelo processador.
- **Linguagem de alto nível:** código textual que permite a codificação de instruções em um nível abstrato elevado mais próximo à linguagem humana.

Linguagem usada nesse curso

- Aqui usaremos a **Linguagem C**, padrão **C99**, definida em 1999 e adotada como padrão ANSI no ano 2000.
- Exemplo de um programa em linguagem C:

```
1  #include <stdio.h>
2
3  /* Este programa lê dois números inteiros,
4  calcula a média entre eles e a apresenta com
5  a precisão de 2 casas decimais. */
6
7  int main() {
8      int a, b;
9      float c;
10     scanf("%d%d", &a, &b);
11     c = (a+b)/2.0;
12     printf("Média: %.2f\n",c);
13     return 0;
14 }
```

Compilação de um programa em C

Linha de comando para a compilação: **gcc -o p1 main.c**

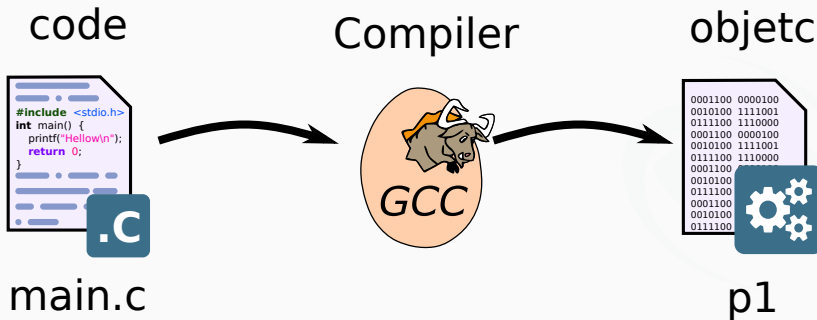


Figura 3: Processo de compilação de um programa em linguagem C.

Estruturas Básicas de uma Linguagem de Programação

Estruturas básicas de uma Linguagem de Programação

1. Declaração de variáveis
2. Comando de entrada de dados
3. Comando de saída de dados
4. Comando de atribuição
5. Expressões (aritméticas e lógicas)
6. Comando de Decisão/Seleção/Desvio de código
7. Comando de Repetição

Prof. Dr. Thiersen Couto Rosa (R)

Exemplo: Fatorial de um número

Implemente um programa que leia um número inteiro, calcule e apresente o seu fatorial.

Sabemos que:

- O fatorial de um número é dado por:

$$n! = 1 \times 2 \times 3 \times \dots \times (n-1) \times n = \prod_{x=1}^n x \quad (2)$$

- Se $n = 0$, $n! = 1$
- Se $n < 0$, o fatorial não existe

Estruturas presentes no código C para cálculo do fatorial

```
1  #include <stdio.h>
2
3  int main() {
4      int n;
5      unsigned long int f;
6
7      f = 1;
8      scanf("%d", &n);
9
10     if(n<0) {
11         printf("n não pode ser negativo.\n");
12         return 0;
13     } else {
14         // Calcula o fatorial de n = n*(n-1)*(n-2)*...*1
15         while( n > 0 ) {
16             f *= n;
17             n = n-1;      // decrementa o valor de n
18         }
19         printf("Fatorial de %u é %lu\n",n,f);
20     }
21     return 0;
22 }
```




H. Farrer.

Algoritmos Estruturados.

LTC, 1999.



T. Gostoso.

Bolo simples.

<http://www.tudogostoso.com.br/receita/29124-bolo-simples.html>, 2017.

2017-03-16.



Wikipedia.

Algoritmo.

<https://pt.wikipedia.org/wiki/Algoritmo>, 2017.

2017-03-16.



Wikipedia.

Tabela ascii.

[https://commons.wikimedia.org/wiki/File:
ASCII-Table-wide.svg](https://commons.wikimedia.org/wiki/File:ASCII-Table-wide.svg), 2018.
2018-03-19.