

# Computação eletrônica: Introdução

**Gurvan Huiban**  
ghuiban@cin.ufpe.br

01 de abril de 2014

# Plano de aula

- 1 Ementa
- 2 Computadores e algoritmos
- 3 A linguagem C

# Computação eletrônica

## Objetivo

Conceitos básicos e prática de programação usando a linguagem C

## Conteúdo programático

- Variáveis, tipos de Dados
- Expressões (aritméticas, lógicas e relacionais)
- Funções pré-definidas, estrutura de programa
- Comandos de entrada/saída, comando de atribuição
- Estruturas condicionais (comandos *if*, *else*, *switch*)
- Estruturas de repetição (comandos *for*, *while*, *do*)
- Subprogramas
- Vetores e matrizes (arrays uni- e bidimensionais)
- Estruturas
- Ponteiros, alocação dinâmica de memória
- Arquivos

# Avaliação

## Metodologia de ensino

- Aulas teóricas
- Aulas práticas  
(Exercícios em sala e no laboratório)
- Listas de exercícios e Monitores

## Avaliação

- 3 Provas  
Média = (Prova 1 + Prova 2 + Prova 3)/3
- Página do curso, com calendário

<http://cin.ufpe.br/~ghuiban/teaching/if165/index.html>

## Referências bibliográficas

- *C: Como Programar, 6ª ed.*, Paul Deitel et al. Editora Makron Books, 2011.
- *Aprendendo a Programar Programando na Linguagem C, 3ª ed.*, Jaime Evarista <http://professor.ic.ufal.br/jaime/>.
- *Treinamento em Linguagem C: módulo 1*, Victorine V. Mizrahi. Editora McGraw-Hill, 1995.
- *Treinamento em Linguagem C: módulo 2*, Victorine V. Mizrahi. Editora McGraw-Hill, 1995.
- *A Book on C, 4ª ed.*, Al Kelley and Ira Pohl. Editora Addison Wesley, 1998.
- *C - completo e total. 3ª ed.*, H. Schildt. Editora Makron Books, 2005.
- *Learn C/C++ today*, V. Carpenter.  
<http://www.cyberdiem.com/vin/learn.html>: Uma coleção de referências e tutoriais sobre as linguagens C e C++ disponíveis na Internet.

# Ambiente de Desenvolvimento - IDE

- **Code::Blocks** - <http://www.codeblocks.org>
- **MinGW** - <http://www.mingw.org/>
- **Dev C++** -  
<http://sf.net/projects/orwelldevcpp/>

- 1 Ementa
- 2 **Computadores e algoritmos**
  - O computador
  - Algoritmos
  - Organização de um computador
  - Linguagens de programação
- 3 A linguagem C



# O computador

Um computador é uma máquina que pode mudar de comportamento:

- Tocar música
- Comunicar
- Gerar documentos
- Simular o funcionamento de um sistema
- Calcular soluções de problema
- ...

# Como fazer?

Como modificar o comportamento do computador?

**Programando o computador.**

# Por que?

## Importância da programação

- Aproveitar do poder oferecido pelo computador sem depender de um software em particular
- Escrever software correspondendo às suas exigências
- Alguns softwares são sofisticados e exigem conhecimentos em programação:
  - Automação de tarefas
  - Excel, Word
  - Software matemáticos
  - ...

# Algoritmos

## Como se programa um computador?

Através de uma sequência de comandos chamada de **algoritmo**.


## Um algoritmo é

- Um conjunto finito de comandos
- Bem definidos (não ambíguos)
- Que soluciona um problema (saída)
- Á partir de dados (entrada)
- Em um tempo finito.

## Bœuf bourguignon!<sup>1</sup>

Limpe a peça de carne, retirando os nervos. Corte em cubos. Tempere com sal e pimenta-do-reino. Coloque o óleo em uma panela grande e leve ao fogo. Quando ferver, doure a carne. Junte a cebola, a cenoura e o alho e refogue mais um pouco. Polvilhe com a farinha e deixe cozinhar em fogo baixo por mais 5 minutos. Acrescente o extrato de tomate, o vinho tinto e o *bouquet garni*. Cozinhe por mais 10 minutos. Adicione o caldo de carne, tampe a panela e deixe cozinhar até a carne ficar macia, em fogo baixo. Quando a carne estiver cozida, separe os cubos de carne e o *bouquet garni*. Passe o molho pela peneira. Leve o molho de volta à panela e junte os cubos de carne.

---

<sup>1</sup>Receita adaptada do blog CLGourmet do site CLmais 

# Linguagem

Entretanto, um computador não consegue compreender uma receita escrita livremente.

Um computador trabalha com uma linguagem mais precisa:

- **Comandos** bem definidos
- **Sequência** bem definida
- Testes para definir qual o próximo passo a ser executado (**se-então-senão**)
- Repetição de comandos até que uma condição seja satisfeita (**enquanto**)

## Bœuf bourguignon: algoritmo

- 1 Se a carne tiver nervos então
  - 1 Limpe a carne.
- 2 Corte em cubos.
- 3 Tempere com sal e pimenta-do-reino.
- 4 Coloque o óleo em uma panela grande.
- 5 Leve ao fogo.
- 6 Espere até ferver.
- 7 Doure a carne.
- 8 Junte a cebola, a cenoura e o alho.
- 9 Refogue mais um pouco.
- 10 Polvilhe com a farinha.
- 11 Deixe cozinhar em fogo baixo por mais 5 minutos.

## Bœuf bourguignon: algoritmo (2)

- 1 Acrescente o extrato de tomate, o vinho tinto e o *bouquet garni*.
- 2 Cozinhe por mais 10 minutos.
- 3 Adicione o caldo de carne.
- 4 Tampe a panela.
- 5 **Enquanto** a carne não estiver macia
  - 1 Deixe cozinhar em fogo baixo.
- 6 **Se** A carne estiver cozida **então**
  - 1 Separe os cubos de carne e o *bouquet garni*.
- 7 Passe o molho pela peneira.
- 8 Leve o molho de volta à panela.
- 9 Junte os cubos de carne.



# Exercício

Consideremos a seguinte questão: como obter exatamente 4 litros de água dispondo de dois recipientes com capacidades de 3 litros e 5 litros?

Como  $4 = 3 + 1$  ou  $4 = 5 - 1$  conseguiremos resolver a questão se conseguirmos obter 1 litro.

Mas isto é fácil, pois  $1 = 3 + 3 - 5!$

- 1 Encha o recipiente de 3 litros.
- 2 Transfira o conteúdo do recipiente de 3 litros para o recipiente de 5 litros.
- 3 Encha o recipiente de 3 litros.
- 4 Com o conteúdo do recipiente de 3 litros, complete o recipiente de 5 litros.
- 5 Esvazie o recipiente de 5 litros.
- 6 Transfira o conteúdo do recipiente de três litros para o recipiente de 5 litros.
- 7 Encha o recipiente de 3 litros.
- 8 Transfira o conteúdo do recipiente de 3 litros para o recipiente de 5 litros.

# Explicação:

Para compreender o algoritmo, sejam  $A$  e  $B$  os recipientes de 3 litros e de 5 litros, respectivamente, e indiquemos por  $(X, n)$  o fato de o recipiente  $X$  conter  $n$  litros de água.

No início temos  $(A, 0)$  e  $(B, 0)$  e, após a execução de cada instrução, teremos:

- 1  $(A, 3), (B, 0);$
- 2  $(A, 0), (B, 3);$
- 3  $(A, 3), (B, 3);$
- 4  $(A, 1), (B, 5);$
- 5  $(A, 1), (B, 0);$
- 6  $(A, 0), (B, 1);$
- 7  $(A, 3), (B, 1);$
- 8  $(A, 0), (B, 4).$

# Outras soluções?

Significa que, se conseguirmos tirar 1 litro do recipiente de 5 litros quando ele estiver cheio, resolveremos a questão.

Para conseguir isto, basta que o recipiente de 3 litros contenha 2 litros.

E para se obter 2 litros? Aí basta ver que  $2 = 5 - 3$ .

## Outra solução: Algoritmo

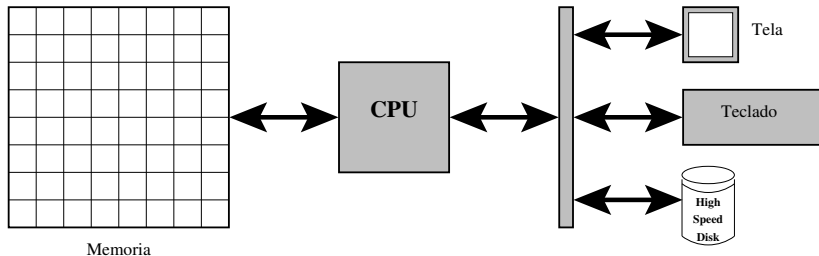
- 1 Encha o recipiente de 5 litros.
- 2 Com o conteúdo do recipiente de 5 litros, encha o de 3 litros.
- 3 Esvazie o recipiente de 3 litros.
- 4 Transfira o conteúdo do recipiente de 5 litros para o recipiente de 3 litros.
- 5 Encha o recipiente de 5 litros.
- 6 Com o conteúdo do recipiente de 5 litros, complete o recipiente de 3 litros.

# Exercício

Um senhor está numa das margens de um rio com uma raposa, uma galinha e um saco de milho. O senhor pretende atravessar o rio com suas cargas, num barco que só comporta o senhor e 1 das cargas. O senhor não pode deixar em uma das margens, sozinhos, a raposa e a galinha, nem a galinha e o milho.

Oriente o senhor a realizar o transporte.

# Organização de um computador





# CPU/Memória

## CPU (*Central Processing Unit*)

- Unidade de Processamento Central
- “Cérebro” do computador
- Coordenação das atividades de entrada e saída
- Coordenação do armazenamento em memória
- Realização dos processamentos

## Memória RAM (*Random Access Memory*)

- Memória de acesso aleatório
- Armazenamento das informações utilizadas no processamento
- Armazenamento os programas

# Entradas/saídas

- Interações com o usuário
- Apresentação dos resultados
- Fonte de dados

## Exemplos

- Teclado
- Mouse
- Tela
- Disco
- Impressora
- Placa de rede
- ...

# Representação da informação

## bit (*binary digit*)

- 0 ou 1
- magnetizado ou não
- corrente eléctrica passa/não passa

## Comunicação com o usuário

Símbolos codificados em sequência de bit

Ex: ASCII: 1 byte = 8 bits: um caractere

'a' : 01100001

# Conjunto de instruções

- Conjunto de instruções que a CPU entende é limitado
- Dependência da arquitetura do computador
- Binário: pouco prático

# Linguagens de programação

## Linguagem de alto nível

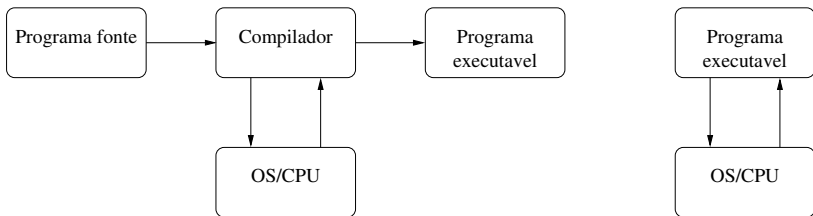
- Usam caracteres, palavras e expressões de um idioma
- Mais fácil escrever e entender
- Ex. C/C++, Pascal, Basic, Java, HTML, ...

# Compilador

## Tradutor

- Programas escritos em uma linguagem de programação
- Para programas em linguagem de máquina

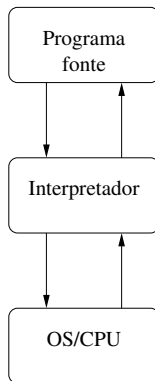
Uma vez convertido, a execução não depende mais do compilador ou do código fonte.



# Interpretador

O interpretador é um programa que traduz em tempo real os programas escritos em alguma linguagem de programação, e manda executar.

A execução de um programa interpretado é em geral mais lenta que o programa compilado.



# A linguagem C

- 1 Ementa
- 2 Computadores e algoritmos
- 3 A linguagem C**



# A origem de C

## Histórico

- Criada por Dennis Ritchie nos anos setenta
- Linguagem do sistema Unix
- Padrão descrito em:  
*The C Programming Language*, Brian Kernighan & Dennis Ritchie
- Grande número de compiladores C com a popularização dos PCs
- Standard C, ANSI (American National Standards Institute)
- Inicialmente: programação de sistemas

# Características

- Linguagem compilada
- Simplicidade: 32 palavras chaves
- Eficiente
- Portável
- Grande variedade de tipos de dados (inteiro, real, array etc.)
- Linguagem estruturada

# Linguagem Estruturada

- Agrupamento de informações e instruções necessárias à uma tarefa específica
- Uso de sub-rotinas (funções) e variáveis locais
- Compartilhamento de uma função entre diferentes seções do código
- Implementação escondida do resto do programa

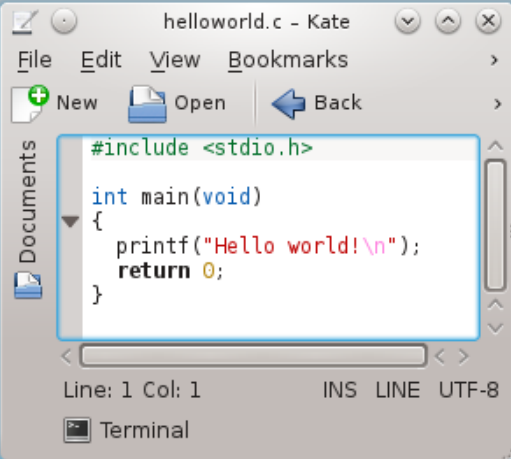
# Por que o C?

- Muito usada
- Eficiência
- Portabilidade
- Compiladores na grande maioria das arquiteturas
- Se encontra em todo tipo de sistema  
(sistemas embutidos até grandes sistemas)
- Sintaxe aparece em outras linguagens  
(C++, Java, PHP, ...)
- Existem bibliotecas em várias áreas

# Hello World

```
int main(void)
{
    printf("Hello world!\n");
    return 0;
}
```

# Hello World: Edição



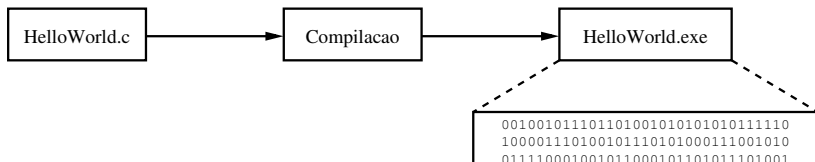
The image shows a screenshot of the Kate text editor window. The title bar reads "helloworld.c - Kate". The menu bar includes "File", "Edit", "View", and "Bookmarks". The toolbar contains icons for "New", "Open", and "Back". The main text area contains the following C code:

```
#include <stdio.h>

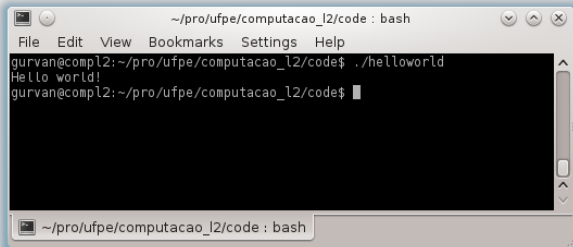
int main(void)
{
    printf("Hello world!\n");
    return 0;
}
```

The status bar at the bottom indicates "Line: 1 Col: 1", "INS LINE", and "UTF-8". A "Terminal" icon is visible at the bottom left of the editor window.

# Hello World: Compilação



# Hello World: Execução



A terminal window titled `~/pro/ufpe/computacao_l2/code : bash` with a menu bar containing `File Edit View Bookmarks Settings Help`. The terminal output shows the command `./helloworld` being executed, resulting in the output `Hello world!`. The prompt `gurvan@comp12: ~/pro/ufpe/computacao_l2/code$` is visible before and after the command. A scrollbar is visible on the right side of the terminal area.

```
~/pro/ufpe/computacao_l2/code : bash
File Edit View Bookmarks Settings Help
gurvan@comp12: ~/pro/ufpe/computacao_l2/code$ ./helloworld
Hello world!
gurvan@comp12: ~/pro/ufpe/computacao_l2/code$
```