

Apresentação

Linguagem de programação

Linguagem de programação

- Estamos em um curso de **Programação**
- Estudaremos algoritmos utilizando uma linguagem específica
 - linguagem **C**
- Mas, o que são algoritmos?
 - Sequência finita e não ambígua de passos para a solução de um problema.
 - Lembre-se que:
 - Diferentes algoritmos podem levar a solução de um mesmo problema



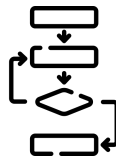
Como preparar um omelete?

- Se você souber fazer, provavelmente não nota mais os passos que realiza:

1. Pegar os ovos
2. Quebrar os ovos
3. Bater os ovos
4. Escolher o recheio
5. Cozinhar
6. Retirar do fogo
7. Servir num prato

Qual o problema?

- Bom, então eu sei fazer algoritmos!!!
- Basta sequenciar tudo que vou fazer para achar uma solução
 - Sim, passa por aí!
 - Inclua lógica para as tomadas de decisões
- Mas qual o problema então em se programar?
 - A língua portuguesa (assim como qualquer outra) é extremamente extensa e pode ser ambígua. **Imagine um computador para interpretar ela.**
 - **Precisamos de outra forma de expressar algoritmos**



Formas de representação

- **Linguagem Natural**: Os algoritmos são expressos diretamente em linguagem natural (o português, por exemplo).
- **Fluxograma**
- **Pseudocódigo**
- **Linguagem de Programação**: Método padronizado para expressar instruções para um computador. É um conjunto de regras sintáticas e semânticas usadas para definir um programa de computador.



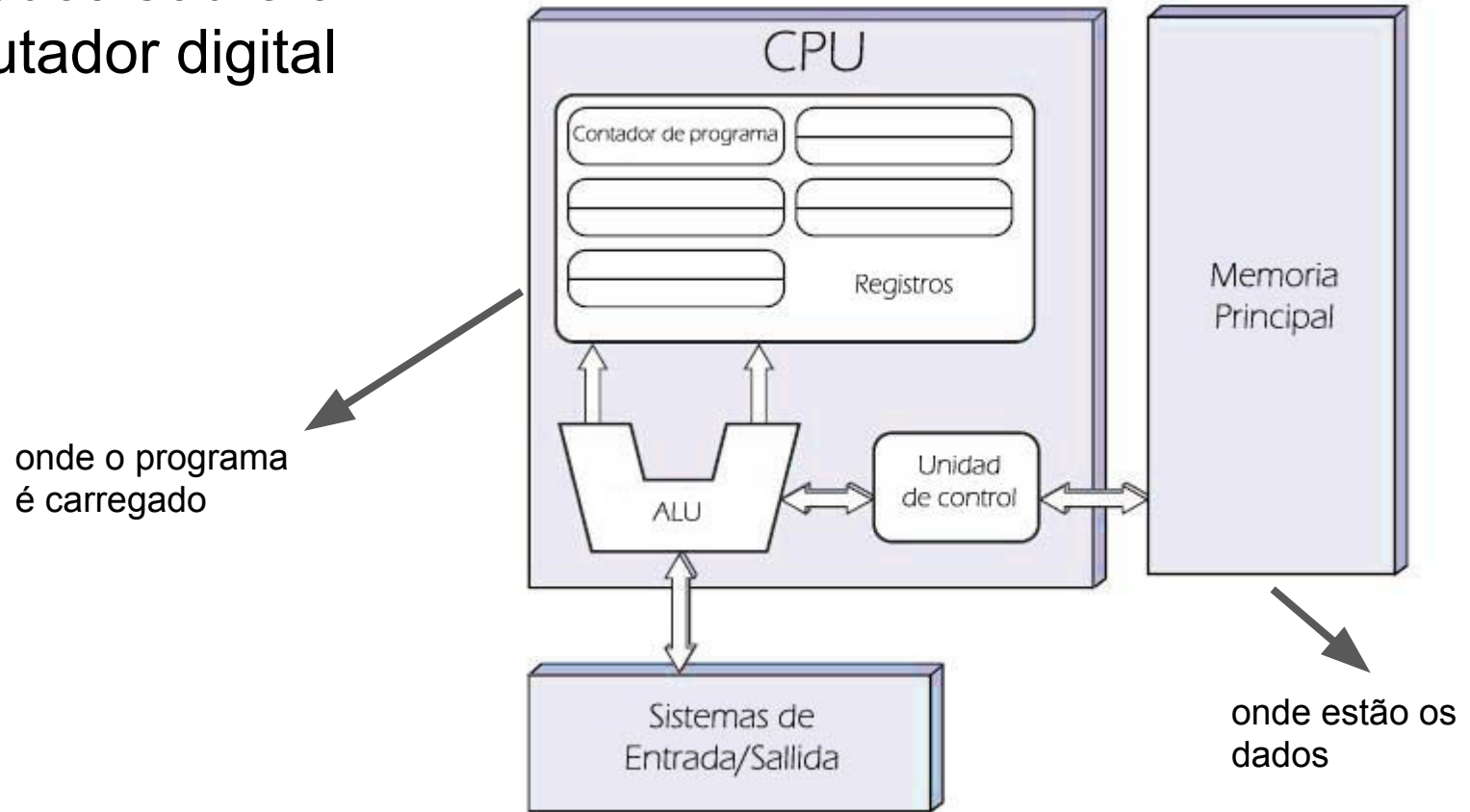
Linguagem de programação

- Possui gramática e significado bem definidos
- Deve ser implementável (executável) com eficiência “aceitável”
- Universal: deve ser possível expressar todo problema computável
- Natural para expressar problemas (em um certo domínio de aplicação)

Mas para isso,



Um pouco sobre o computador digital



Linguagem de máquina

- O computador (ou máquina) executa instruções bem definidas, contidas em uma linguagem de máquina
 - é a linguagem de mais baixo nível de entendimento para o ser humano;
 - e a única, entendida pelo processador.
- constituída inteiramente de números (código binário)
 - o que torna muito difícil de entendê-la
- Cada processador tem seu conjunto único de instruções, que definem sua linguagem de máquina, estabelecido pelo fabricante do chip.
- Uma instrução típica de linguagem de máquina seria algo como: **0100 1111 1010**

01100
10110
11110

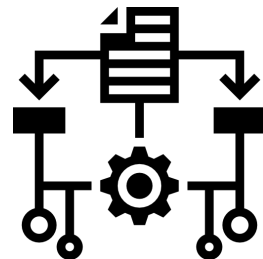
Linguagem de máquina

- Como podemos nos comunicar com o computador, utilizando uma linguagem tão difícil?
 - não podemos...
- Para isso, foram criadas as linguagens de programação
- As linguagens de programação representam abstrações que são convertidas para linguagem de máquina
 - Por um compilador
 - Ou por um interpretador

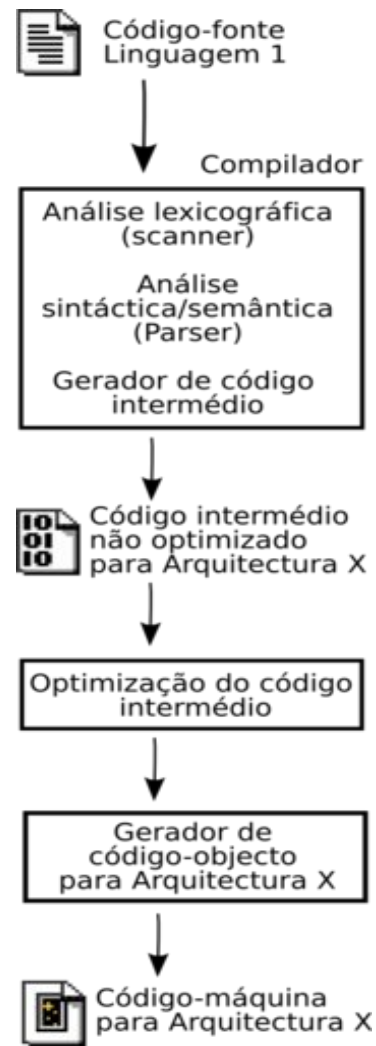
10101
01011
10101

Compilador

- A partir de um **código-fonte** escrito em uma linguagem, cria um programa equivalente, mas escrito na **linguagem objeto (máquina)**
- Composto por:
 - Analisador Léxico
 - Verifica a compatibilidade com determinado alfabeto
 - Analisador Sintático
 - Verifica a estrutura gramatical de uma sentença
 - Gerador de Código
 - Otimizador

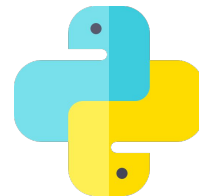
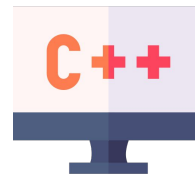


Compilador



E quais linguagens existem?

- Fortran
- C
- C++
- C#
- Java
- Python
- Pascal
- Object Pascal
- LUA
- Visual Basic
-



Por que tantas linguagens?

- Propósitos diferentes
- Avanços tecnológicos
- Interesses comerciais
- Cultura e background científico

Sobre o C

Linguagem de Programação

As Origens de C



Ken Thompson e Dennis Ritchie (da esquerda pra direita), os criadores das linguagens **B** e **C**, respectivamente.

História do C

- O C foi desenvolvido a partir de duas linguagens anteriores, o BCPL e o B.
- O BCPL foi desenvolvido em 1967 por Martin Richards
- **Ken Thompson** criou a linguagem B com base no BCPL e usou o B para criar as primeiras versões do sistema operacional UNIX
- Tanto o BCPL como o B eram linguagens "sem tipos" ("typeless")
 - a responsabilidade de lidar com números inteiros ou reais, por exemplo, recaía sobre os ombros do programador.
- A linguagem C foi desenvolvida a partir do B por Dennis Ritchie

História do C

- Hoje em dia, praticamente todos os grandes sistemas operacionais estão escritos em C e/ou C++
- O C usa muitos dos importantes conceitos do BCPL e do B, além de adicionar **tipos de dados**
- A rápida expansão do C em vários tipos de computadores levou a muitas variantes
 - Elas eram similares, mas freqüentemente incompatíveis
 - Isso se tornou um problema sério, pois precisava-se que um código fosse executado em diferentes plataformas
- Então, começando em 1983, o comitê ANSI X3J11 padronizou a linguagem
 - esse padrão chama-se **ANSI C**

Por que aprender C?

- Imagine que você comprou um carro com câmbio automático; você dirige ele por um tempo (ilegalmente) e vai fazer um exame, com carro manual, para obter uma CNH
- Você nunca dirigiu um carro manual antes
- Provavelmente você irá falhar no exame
- Se você tivesse aprendido a dirigir um carro manual antes, a versão automática seria muito mais fácil de aprender
- O efeito de aprender C é semelhante:
 - Aprender C, te torna capaz de aprender facilmente qualquer outra linguagem moderna

Vantagens do C

- **C é uma linguagem de nível médio:**
 - Está entre as linguagens de baixo-nível (compreensíveis por máquinas), e as de alto-nível (mais amigáveis ao usuário)
 - Pode ser usado para escrever sistemas operacionais (como Windows e Linux) e também para programar em nível de aplicativo
 - Estudar C ajuda no entendimento da arquitetura do computador e conceitos como:
 - ponteiros e localização de memória

Vantagens do C:

- **Ajuda a entender os fundamentos das teorias de computador**
 - áreas como Redes de Computadores, Design de Compiladores, Arquitetura de Computadores e Sistemas Operacionais são baseadas em C, e requerem um bom conhecimento nessa linguagem
 - Nas linguagens mais modernas, os detalhes no nível da máquina são ocultados ao usuário
 - Então, para trabalhar com cache da CPU, memória, adaptadores de rede, aprender a programação C é uma obrigação.

Vantagens do C:

- **Menos bibliotecas:**

- O C possui menos bibliotecas em comparação com outras linguagens de alto nível.
 - Portanto, aprender C também ajuda a fixar conceitos de programação, pois você precisa escrever muitas coisas do zero.
- Você não dependerá inteiramente da LP para implementar algumas operações básicas
- Implementações por conta própria também o ajudarão a desenvolver suas habilidades analíticas.
 - como fazer a execução do código ser mais rápida
 - como evitar uso desnecessário de Memória RAM

Vantagens do C

- **Muito rápida em termos de execução**

- C executa muito mais rápido quando comparado a qualquer outra linguagem de programação
- Devido a não existência de alguns procedimentos automáticos executados pela maioria das linguagens. Em C, esse procedimentos ficam por **responsabilidade do programador!**
- Por exemplo, em JAVA:
 - Há checagem de índices *out of bounds* do vetor
 - índice maior que o tamanho do vetor
 - checagem de vazamentos de memória (*memory leaks*)
 - quando se perde a referência de uma parte da memória previamente alocada
 - Exception handling
 - garbage collection
 - gerenciamento de memória, livrando-se de espaço de memória que não será mais utilizado

Vantagens do C

- **Programação embarcada:**
 - C é amplamente usado em programação embarcada
 - Na programação de microcontroladores e microprocessadores para controlar seus periféricos e manipular circuitos externos;
 - utilizados nas áreas de automóveis, robótica, hardware, etc.

Estrutura geral de um programa em C

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

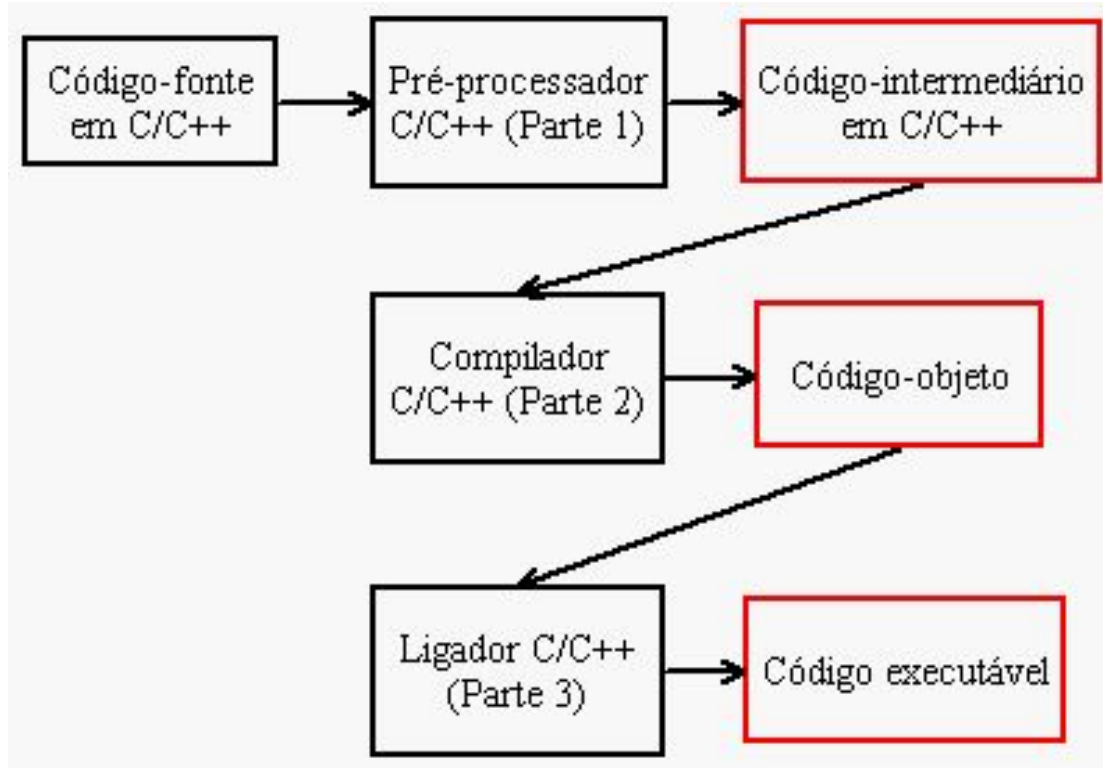
```
int main() {  
    printf("Hello World\n");  
    return 0;  
}
```

- todo programa em C deve possuir pelo menos 1 função, a **main()**
 - Função a ser chamada quando a execução começa
 - Com retorno e passagem de parâmetro

Edição, compilação e Link-edição

- Gerar um programa executável em C consiste neste três passos:
 - Criar o programa
 - Compilar o programa
 - Link-editar o programa com as funções necessárias da biblioteca

Edição, compilação e Link-edição



Alguns conceitos ...

- **Código-fonte:** o texto de um programa que um usuário pode ler, normalmente interpretado como o programa. Entrada do compilador.
- **Código-objeto:** tradução do código-fonte de um programa em código de máquina que o computador pode ler e executar diretamente (arquivo objeto). Entrada para o link-editor.
 - Cada arquivo.c gera um arquivo.o (objeto)
- **Link-editor:** um programa que une bibliotecas e arquivos objetos compilados separadamente, em um único programa. A saída é um programa executável.

Alguns conceitos...

- **Biblioteca:** arquivo contendo as funções padrão que seu programa pode usar. Geralmente com a notação *nome_biblioteca.h*
- **Tempo de compilação:** eventos que ocorrem enquanto seu programa está sendo compilado. Ex.: erro de sintaxe.
- **Tempo de execução:** eventos que ocorrem enquanto o seu programa é executado.

Avaliação

- N1 = prova 1
- N2 = prova 2
- N3 = questionários online

$$\text{Nota Final} = (N1+N2+N3) / 3$$