COMPILAÇÃO

Ricardo José Cabeça de Souza

www.ricardojcsouza.com.br

ricardo.souza@ifpa.edu.br

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PARÁ

www.ricardojcsouza.com.br
ricardo.souza@ifpa.edu.br

Compilação

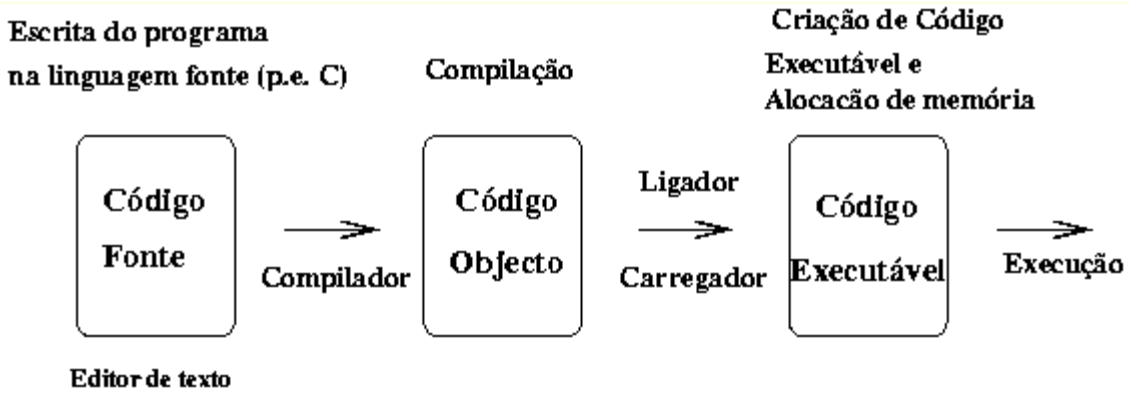
- **Programas**
  - **Código-fonte** escrito em linguagem de programação de alto nível, ou seja, com um nível de abstração muito grande, mais próximo do modo "humano" de se pensar
  - **Código executável** que é um código feito numa linguagem de baixo nível de abstração, muito mais próxima do modo de funcionamento das máquinas do que do raciocínio humano
- Existem duas formas de se transformar um programa escrito em uma linguagem de alto nível em um código capaz de ser executado por um dispositivo computacional
  - Compilador
  - Interpretador

www.ricardojcsouza.com.br
ricardo.souza@ifpa.edu.br

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA PARÁ

Compilação

- **Compilação**
  - O programa escrito na linguagem fonte é traduzido para linguagem máquina e depois ligado e carregado para ser executado



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PARÁ

www.ricardojcsouza.com.br
ricardo.souza@ifpa.edu.br

Compilação

- **Interpretação**
  - O programa fonte é traduzido e executado instrução a instrução, de modo interativo
  - O **Interpretador** traduz cada instrução para uma representação interna e interpreta-a simulando o funcionamento do processador
  - O interpretador aceita para além das instruções da linguagem, comandos para controlar o seu funcionamento

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PARÁ

www.ricardojcsouza.com.br
ricardo.souza@ifpa.edu.br

Compilação

- **PROCESSO DE COMPILAÇÃO**
  - Processo de tradução de um código fonte escrito normalmente em uma linguagem de alto nível (de fácil entendimento por parte do programador) para uma linguagem de baixo nível
  - Um compilador é um programa responsável por executar os processos de compilação

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PARÁ

www.ricardojcsouza.com.br
ricardo.souza@ifpa.edu.br

Compilação

- **FASES DA COMPILAÇÃO**
  - **Análise Léxica**
    - Primeira etapa da compilação
    - A função do analisador léxico, também conhecido como scanner, é analisar todo o código fonte e produzir símbolos (tokens) que podem ser manipulados na etapa seguinte
    - Nesta etapa são eliminados os espaços em branco e comentários
    - Reconhecer as sequências de símbolos que representam uma unidade. P.e. o nome de uma variável, uma constante, uma palavra chave de uma instrução (while)

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PARÁ

www.ricardojcsouza.com.br
ricardo.souza@ifpa.edu.br

Compilação

- **FASES DA COMPILAÇÃO**
  - **Análise Sintática**
    - O analisador sintático (parsing) é quem dá significado às sequências de tokens criadas anteriormente
    - Identifica a estrutura gramatical do programa e reconhece o papel de cada componente
    - É normalmente construída uma *árvore sintática* do programa e uma tabela de símbolos, que identifica variáveis

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PARÁ

www.ricardojcsouza.com.br
ricardo.souza@ifpa.edu.br

Compilação

- **FASES DA COMPILAÇÃO**
  - **Geração do Código**
    - A geração de código pode englobar
      - Análise semântica
      - Geração de código intermediário
      - Otimizadores
      - Geração de código final

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PARÁ

www.ricardojcsouza.com.br
ricardo.souza@ifpa.edu.br

Compilação

- **FASES DA COMPILAÇÃO**
  - **Análise Semântica**
    - Esta etapa é responsável por analisar a semântica, ou significado, de cada elemento do código
    - É ele quem encontra erros como, por exemplo, uma multiplicação entre tipos de dados diferentes

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PARÁ

www.ricardojcsouza.com.br
ricardo.souza@ifpa.edu.br

Compilação

- **FASES DA COMPILAÇÃO**
  - **Geração do Código Intermediário**
    - Nesta etapa ocorre a conversão da árvore sintática, criada na etapa 2, em uma representação intermediário do código fonte
    - Processo de construir instruções da linguagem máquina (em assembly, normalmente) que simulam as instruções reconhecidas pelo analisador sintático

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PARÁ

www.ricardojcsouza.com.br
ricardo.souza@ifpa.edu.br

Compilação

- **FASES DA COMPILAÇÃO**
  - **Otimização do Código**
    - Nesta etapa o código é otimizado para uma determinada arquitetura (hardware e sistema operacional específico)

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PARÁ

www.ricardojcsouza.com.br
ricardo.souza@ifpa.edu.br

Compilação

- **FASES DA COMPILAÇÃO**
  - **Geração de Código Final**
    - Nesta ultima etapa da compilação, o arquivo executável (.exe) é criado, otimizado para aquela arquitetura

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PARÁ

www.ricardojcsouza.com.br
ricardo.souza@ifpa.edu.br

Compilação

- **Compilação C++**
  - Quando invocamos a compilação na nossa **IDE(Integrated Development Environment)** para cada arquivo C/C++ (arquivos .c, cpp, cc, etc) é executado o **pré-processador** (que executa as diretivas #ifdef, #include, substituições de macros, etc), e o resultado disso é um arquivo temporário com o código fonte pré-processado
  - Em seguida é invocado o **compilador** que processa este arquivo, caso não ocorram erros é gerado **o arquivo objeto** (geralmente extensão obj ou .o)
  - Após a geração de todos arquivos objetos é invocado o **linker**, que faz a linkagem de todos os arquivos objeto gerando finalmente o **executável** (exe no caso do windows) ou biblioteca dinâmica (dlls no Windows)

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PARÁ

www.ricardojcsouza.com.br
ricardo.souza@ifpa.edu.br

Compilação

- **Compilação C++**
  - Envolve três passos principais
    - Pré-processador
    - Compilação
    - Linkagem

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PARÁ

www.ricardojcsouza.com.br
ricardo.souza@ifpa.edu.br

Compilação

- **Pré Processador**
  - O pré-processador é a primeira etapa da compilação, pode ser ou não um programa separado do compilador, mas o que importa mesmo é que ele é o primeiro a “tocar” o código e cuida de algumas tarefas como:
    - Junta linhas que foram separadas por sequências de escape.
    - Remove comentários e os substitui por espaços em branco
    - Expande macros
    - Processa diretivas de pré-processamento

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PARÁ

www.ricardojcsouza.com.br
ricardo.souza@ifpa.edu.br

Compilação

- **Pré Processador**
  - O comando de pré-processamento que provavelmente é mais conhecido pelos desenvolvedores é o **include**, que é resolvido pelo pré-processador
  - O include consiste em apenas abrir o arquivo indicado, copiar o seu conteúdo e colar em cima do comando include, sendo que o include pode ter duas formas:
    - #include <iostream>
    - #include “objeto.h”
      - Na primeira forma, usando-se < e > o pré-processador procura pelo arquivo indicado apenas nos diretórios de include que o compilador estiver configurado
      - Na segunda opção com aspas, o pré-processador procura pelo arquivo no diretório onde estão os fontes do programa
  - Vale lembrar que arquivos de header (extensões h, hpp, etc) não são compilados diretamente pelo compilador, mas apenas quando são incluídos em algum arquivo de código (c, cpp, cc, etc).

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PARÁ

www.ricardojcsouza.com.br
ricardo.souza@ifpa.edu.br

Compilação

- **Compilador C/C++**
  - O compilador processa o resultado do pré-processador e para cada unidade compilação gera um **arquivo objeto**
  - Trata cada arquivo fonte como sendo uma unidade de compilação
    - Compila cada um de maneira independente, ignorando a existência de qualquer outro arquivo
    - A única forma do compilador considerar um outro arquivo é através da diretiva **include**, que já foi processada no passo anterior pelo pré-processador
  - O construtor do compilador foca seus esforços na principal tarefa que consiste em verificar se o código esta bem construído segundo a gramática da linguagem
  - Após verificar que não existe qualquer erro sintático (como, por exemplo, um “;” faltando) ou semântico (como, por exemplo, uma variável definida duas vezes) ele já pode então gerar o código em linguagem de maquina que resulta no arquivo **objeto**

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PARÁ

www.ricardojcsouza.com.br
ricardo.souza@ifpa.edu.br

Compilação

- **Linker**
  - O linker ou programa de ligações é responsável por juntar todos os arquivos objetos e gerar o arquivo executável (pode ser usado também para gerar bibliotecas dinâmicas, como dlls)
  - Dessa forma a tarefa dele é substituir todos as chamadas de funções e acessos a variáveis em arquivos objeto pelo endereço real de onde esse item se encontra
  - Fica responsável também por organizar cada função, variável dentro do espaço de memória do executável

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PARÁ

www.ricardojcsouza.com.br
ricardo.souza@ifpa.edu.br

Compilação