

Programa de Ensino

1) Identificação

Disciplina: INE5426 - Construção de Compiladores
Carga horária: 72 horas-aula Teóricas: 20 Práticas: 52
Período: 1º semestre de 2020 até a presente data

2) Cursos

- Ciências da Computação (208)

3) Requisitos

- Ciências da Computação (208)
• INE5421 - Linguagens Formais e Compiladores

4) Ementa

Projeto de especificação de linguagens de programação. Implementação das etapas que compreendem o processo de compilação: Análise Léxica, Análise Sintática, Análise Semântica, Geração e Otimização de Código. Evolução e tendências da área de compiladores e linguagens de programação.

5) Objetivos

Geral: Dotar o aluno de conhecimento básico dos conceitos e técnicas necessários para a construção de compiladores, bem como para a compreensão dos conhecimentos envolvidos no projeto de linguagens de programação e o tratamento computacional de linguagens em geral

Específicos:

- Compreender os aspectos ligados ao projeto de linguagens de programação
- Descrever a organização arquitetural dos compiladores e seu funcionamento
- Compreender e implementar os principais algoritmos de análise léxica.
- Compreender e implementar os principais algoritmos de análise sintática
- Compreender e implementar os processos de análise semântica adotados nos compiladores
- Descrever as técnicas de recuperação de erros utilizadas nos compiladores.
- Identificar as formas de geração e de representação de código intermediário
- Compreender as técnicas de otimização de código e geração de código objeto
- Identificar, avaliar e utilizar ferramentas de apoio na construção de compiladores

6) Conteúdo Programático

- 6.1) A estrutura de um compilador [1 hora-aula]
- 6.2) Linguagens de programação [1 horas-aula]
 - Características principais
- 6.3) Especificação e projeto de uma linguagem [6 horas-aula]
- 6.4) Análise léxica [2 horas-aula]
- 6.5) Construção de um analisador léxico [8 horas-aula]
- 6.6) Análise sintática e correção de erros [6 horas-aula]

- 6.7) Construção de um analisador sintático [12 horas-aula]
- 6.8) Análise semântica [6 horas-aula]
- 6.9) Implementação da análise semântica [12 horas-aula]
- 6.10) Geração de código intermediário e otimização [6 horas-aula]
- 6.11) Implementação do gerador de código [12 horas-aula]

7) Bibliografia Básica

- MOGENSEN, T. AE. Introduction to Compiler Design. Springer, 2011.
- SU, Y., and YAN, S. Y. Principles of Compilers. Springer, 2011.
- BORNAT, R. Understanding and Writing Compilers. Springer, 1979.
- DOS REIS, A. J. Compiler Construction Using Java, JavaCC, and Yacc. Wiley, 2012.

8) Bibliografia Complementar

- AHO, A.V.; LAM, M. S.; SETHI, R. ULLMAN, J.D. Compiladores – Princípios, Técnicas e Ferramentas, Pearson, 2008
- DELAMARO, Márcio Eduardo. Como Construir um acompilador. São Paulo, Novatec, 2004.
- PRICE, Ana Maria de Alencar, TOSCANI, Simão Sirineo. Implementação de Linguagens de Programação: Compiladores. Porto Alegre, Sagra, 2004.
- HOPCROFT, J. E., ULLMAN, J. D. Formal Languages and Their Relations to Automata. Addison-Wesley, 1969..
- HOPCROFT, J. F., ULLMAN, J. D.. Introduction to Automata Theory, Languages and Computation. Ed. Addison-Wesley, 1979
- MENESES, P. B. Linguagens Formais e Autômatos, Ed. Sagra Luzzato, 2. edição, 1998.
- AHO, A. V., ULLMAN, J. D. The Theory of Parsing, Translation, and Compiling. Volume I: Parsing. Ed Prentice-Hall, Inc. 1972, 542p