



PLANO DE ENSINO

1. IDENTIFICAÇÃO

Curso: Ciência da Computação - Noturno

Componente Curricular: Construção de Compiladores

Fase: 7 (sete)

Ano/Semestre: 2016/2

Número da Turma: 14944

Numero de Créditos: 4 (quatro)

Carga horária - Hora Aula: 72 horas

Carga horária - Hora Relógio: 60 horas

Professor: Bráulio Adriano de Mello

Atendimento aos estudantes: Segunda entre 18:00h e 20:00h

2. OBJETIVO GERAL DO CURSO

O curso tem por objetivo a formação integral de novos cientistas e profissionais da computação, os quais deverão possuir conhecimentos técnicos e científicos e serem capazes de aplicar estes conhecimentos, de forma inovadora e transformadora, nas diferentes áreas de conhecimento da Computação. Adicionalmente, os egressos do curso deverão ser capazes de adaptar-se às constantes mudanças tecnológicas e sociais, e ter uma formação ao mesmo tempo cidadã, interdisciplinar e profissional.

3. EMENTA

A disciplina de Construção de Compiladores permite ao estudante compreender conceitos e características do processo de compilação. Oferece melhores condições para avaliar as propriedades de uma linguagem e seus efeitos nos códigos gerados considerando, por exemplo, a relação com chamadas ao sistema (sistemas operativos), desdobramentos dos conceitos de linguagens de programação e computabilidade. Podem ser também enumeradas benefícios indiretos resultantes de conhecimentos em torno do processo de compilação. Entre eles, a disponibilidade de referenciais ou informações adicionais que interferem positivamente nos métodos ou práticas de programação.

4. OBJETIVOS

4.1.GERAL

Compreender a estrutura de compiladores, o processo de compilação no reconhecimento de linguagens e a geração/otimização de código, construção de analisadores léxicos e sintáticos.

4.2.ESPECÍFICOS

- Estudo da Estrutura e Processo de compilação
- Entender e construir analisadores observando formalismos e algoritmos utilizados para os diferentes tipos de máquinas reconhecedoras
- Estudo e experimentação de soluções para verificação semântica e otimização de

- código
- Desenvolver um projeto prático atendendo etapas do processo de compilação incentivando os estudantes a realizar esforços integradores com demais áreas do curso, preferencialmente observando os CCRs em curso no semestre;
 - Fortalecer iniciativas tais como elaboração de novos materiais de apoio e dinâmicas alternativas de atividades buscando melhorar as condições didático-pedagógicas para aprendizado de compiladores.

5. CRONOGRAMA E CONTEÚDOS PROGRAMÁTICOS

ENCONTRO	CONTEÚDO
1	Apresentação da disciplina, conceitos gerais, apreciação do plano de ensino e sistema de avaliação.
2,3	Revisão: linguagens, geradores e reconhecedores, hierarquia, construção e simplificação de gramáticas.
4	Componentes de um Compilador e processo de compilação: Analisadores Léxico, Sintático e Semântico e Gerador de Código. Recuperação de estudos e AvPEA-1
5	Interpretação/compilação, analisadores, geração e otimização de código
6	Construção de analisadores léxicos: autômato finito como máquina reconhecedora. TF1 .
7	Estudo de caso: Lex e yacc. Recuperação de estudos e AvPEA-2
8	Discussão do Projeto 1 : Especificação e implementação de reconhecedor léxico.
9, 10	Análise sintática: Analisadores ascendentes e descendentes, simplificação de GLCs, árvores de derivação (esq., dir.), analisadores LR e LL.
11, 12	Autômatos de Pilha como máquina reconhecedora. Recuperação de estudos e AvPEA-3
13	Análise sintática: precedência de operadores (shift-reduce).
14,15	Implementação de analisador sintático para a gramática de operadores com shift-reduce. TF2. Av. Rec rendimento NP1
16, 17	Análise sintática SLR: construção de itens válidos e conjunto de transições (gramática de operadores). Recuperação de estudos e AvPEA-4
18,19	Análise sintática SLR: construção da tabela de parsing.
20,21	Algoritmo de reconhecimento sintático SLR. Recuperação de estudos e AvPEA-5 .
22,23	Discussão projeto 2 : desenvolvimento de reconhecedor sintático SLR
24	Tradução dirigida por sintaxe
25	Análise Semântica e Código intermediário
26,27	Otimização de código: conceitos, características, otimização por Grafos Acíclicos Dirigidos. AvPEA-6
28	Código objeto e editor de ligação. Av Rec rendimento NP3
29,30,31,32, 33	Desenvolvimento e apresentação dos reconhecedores léxico e sintático conforme descrição dos projetos 1 e 2.

34,35	Apresentação dos projetos 1 e 2
36	Recuperação de rendimento (Avaliação Rec rendimento da NP2 e NP4)

Obs.: As atividades/cronograma dos encontros podem ser ajustados no decorrer do semestre.

6. PROCEDIMENTOS METODOLÓGICOS

Embora a disciplina contemple conteúdo a ser trabalhado em encontros teóricos, maior esforço é dedicado ao estudo do processo de construção de compiladores. Deste modo, serão contempladas aulas expositivas/dialogadas para atendimento de conceitos específicos e atividades práticas no projeto e implementação dos processos conforme a estrutura de compiladores. O perfil aplicado da disciplina aponta para maior ênfase em exercícios práticos, demonstrações, contextualização experimentação e implementação.

7. AVALIAÇÃO DO PROCESSO ENSINO-APRENDIZAGEM

A avaliação do processo ensino-aprendizagem inclui 6 oportunidades distribuídas ao longo do período, podendo excepcionalmente variar. As AvaliaçõesPEA (AvPEA) podem fazer uso de métodos diferenciados, como por exemplo: prova teórica, exercício prático, seminários curtos, elaboração de textos técnicos, entre outros. Serão realizados dois projetos de média duração e complexidade que incluem parte prática (implementação) e escrita (artigo técnico).

Na avaliação dos projetos, 50% da nota é atribuída pela parte escrita e implementação (grupo) e 50% da nota é atribuída mediante avaliação individual (clareza, domínio, conhecimento de detalhes e etapas do desenvolvimento), independente do grupo de trabalho.

A Tabela 1 apresenta a configuração das notas parciais, dos instrumentos de avaliação, recuperação de rendimento e média final:

Tabela 1 – Método de avaliação

Distribuição de notas e pesos para processo de avaliação da oferta de CCR		
Nota Parcial	Instrumentos	Recuperação de rendimento
NP1 = $AVG(AvPEA_1 + AvPEA_2 + AvPEA_3)$	Três (3) avaliações parciais (AvPEA) com mesmo peso para compor a NP	Rec.Rend(NP1): $MAX(NP1)$
NP2 = Nota Projeto 1	Apresentação de trabalho prático (implementação + texto)	Rec.Rend(NP2): $MAX(NP2)$ + Se $NP2 < 6,0$, será oportunizada nova apresentação em encontro imediatamente posterior.
NP3 = $AVG(AvPEA_4 + AvPEA_5 + AvPEA_6)$	Três (3) avaliações parciais (AvPEA) com mesmo peso para compor a NP	Rec.Rend(NP3): $MAX(NP3)$
NP4 = Nota Projeto 2	Apresentação de trabalho prático (implementação + texto)	Rec.Rend(NP4): $MAX(NP4)$ + Se $NP4 < 6,0$, será oportunizada nova apresentação em encontro imediatamente posterior.

$$MF = NP1 \cdot 0,20 + NP2 \cdot 0,20 + NP3 \cdot 0,20 + NP4 \cdot 0,40$$

MAX: Avaliação recuperação rendimento (maior nota); NP: Nota Parcial; AvPEA: Avaliação do processo ensino aprendizagem utilizada para compor NPs.

7.1 RECUPERAÇÃO: NOVAS OPORTUNIDADES DE APRENDIZAGEM E AVALIAÇÃO

Em pelo menos três momentos ao longo do semestre o estudante poderá usufruir de oportunidades de recuperação de rendimento conforme critérios e requisitos

apresentados na Tabela 1. A avaliação de recuperação de rendimento inclui os conteúdos avaliados pelos instrumentos de avaliação que compõe a respectiva NP;

Serão considerados os trabalhos facultativos (TF) individuais que, se entregues conforme data limite a ser definida durante os encontros, podem ser utilizados para aumento da média final limitado a 0,5 ponto num total de 10. Trabalhos não realizados não implicam em redução de nota.

Os TFs facultam o acréscimo de até 0,5 ponto (num total de 10) na média final quando média provisória final ficar entre 5,5 e 5,9. Neste caso, a entrega dos TFs é requisito para ajuste da nota no limite da nota mínima para aprovação. O tópico de cada TF é discutido com os estudantes durante os encontros. O número de TFs também é flexível.

Todas as AvaliaçõesPEA são precedidas de oportunidades para recuperação de estudos e se apresentam, também, como oportunidade de recuperação de rendimento ao dividir o conteúdo em múltiplos momentos de avaliação. O horário extra-classe para atendimento de estudantes também oportuniza a recuperação de estudos.

8. REFERÊNCIAS

8.1 BÁSICA

PRICE, A. M. A., TOSCANI, S. S. Implementação de Linguagens de Programação: Compiladores. Bookman Companhia Ed., 2008.

HOPCROFT, J. F., ULLMAN, J. D., Motwani, R., "Introdução a teoria dos automatos", Ed. Campus, 2002.

AHO, A. V., SETHI, R., LAM, M., "Compiladores: Princípios, técnicas e ferramentas", Ed. Longman do Brasil, 2007.

GRUNE, D., BAL, H. E., JACOBS, C., LANGENDOEN, K. Projeto Moderno De Compiladores: implementação e aplicações. Rio de Janeiro: Campus, 2001.

8.2 COMPLEMENTAR

WOOD, D. , "Theory of Computation", Ed. John Wiley & Sons, 1987.

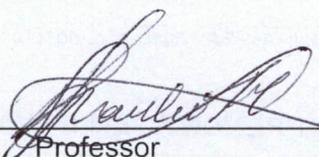
FURTADO, O. J. V., "Apostila de Linguagens Formais e Compiladores – versão 2", UFSC, 2002.

DELAMARO, M. E. **Como Construir Um Compilador Utilizando Ferramentas Java**. Rio de Janeiro: NOVATEC, 2004.

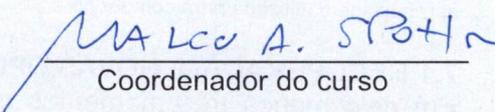
LOUDEN, K. C. **Compiladores Princípios e Práticas**. São Paulo: THOMSON PIONEIRA, 2004.

Furtado, Olinto. Linguagens formais e compiladores. <www.inf.ufsc.br/~olinto/apostila-lfc.doc>. Acesso em 30/09/2012.

8.3 SUGESTÕES



Professor
1645173



Coordenador do curso
MARCO AURÉLIO SPOHN
Siape nº. 1521671
Coord. do Curso de Ciência da Computação
Universidade Federal da Fronteira Sul-UFFS
Campus Chapecó-SC