UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
DEPARTAMENTO DE INFORMÁTICA APLICADA

DISCIPLINA: **COMPILADORES**
CÓDIGO: **INF01147 - INF01033**
CURSOS/PRE-REQUISITO:

CIC	ECP
Linguagens formais e autômatos N	Teoria da Computação

CARGA HORÁRIA: **4**
CRÉDITOS: **4**
Professor Regente: **Nicolas Maillard**
Professor Turma:

SÚMULA

Análise léxica e sintática. Tradução dirigida por sintaxe. Otimização de código. Processadores de linguagens.

OBJETIVOS

Um compilador traduz um texto escrito numa linguagem fonte para uma linguagem alvo, em geral linguagem de máquina. Essa disciplina apresenta as várias etapas necessárias para tal (análise lexical, sintática e semântica, otimização e geração de código) através do estudo dos algoritmos eficientes que são usados e de sua implementação em ferramentas clássicas. Aulas em laboratório possibilitam o estudo prático dos conceitos vistos em aula teórica.

CONTEÚDO PROGRAMÁTICO

1. Apresentação da disciplina. Introdução a Compiladores.
2. Análise lexical - expressões regulares e reconhecedores.
3. Análise sintática - gramáticas livres de contexto, reconhecedores : parsers top-down e bottom-ups ; conjuntos First e Follows ; tabelas preditivas e LR.
4. Análise semântica ; atributos semânticos herdados e sintetizados ; esquemas S e L atribuídos.
5. Geração de código intermediário para atribuição, checagem de tipos, vetores e estruturas de controle de fluxo de execução (laços, testes).
6. Suporte ao run-time - registro de ativação - transferência de parâmetros.
7. Geração de código assembly .
8. Otimização de código - bloco básico - laços naturais.

METODOLOGIA

Haverá aulas durante o semestre. Os alunos poderão assistir às aulas, fazer perguntas e consultar o material disponibilizado on-line pelo professor, além de consultar as referências bibliográficas disponíveis na biblioteca. Haverá trabalhos e provas para verificar o correto entendimento dos conceitos da disciplina.

CRONOGRAMA DE ATIVIDADES:

Observação: O cronograma inclui a suspensão de aulas durante a semana academica (25 a 29/05/2009).

Encontro – Data – Atividade:

01 - 3a, 03 de Marco, 2009

Aula inaugural : definições, requisitos e posicionamento da disciplina.

02 - 5a, 05 de Marco, 2009

Análise lexical (1/2): Expressões Regulares (E.R.) - tradução em autômatos.

03 - 3a, 10 de Marco, 2009

Análise lexical (2/2): Reconhecimento de E.R. - uso de (F)LEX.

04 - 5a, 12 de Marco, 2009

Laboratório - encontro etapa 1 (estruturas de dados).

05 - 3a, 17 de Marco, 2009

Análise sintática (1/5) - Gramáticas Livres de Contexto.

06 - 5a, 19 de Marco, 2009

Análise sintática (2/5): análise top-down - definição de First/Follow.

07 - 3a, 24 de Marco, 2009

Análise sintática (3/5): análise top-down com tabela preditiva.

08 - 5a, 26 de Marco, 2009

Laboratório - encontro etapa 2 (lex).

09 - 3a, 31 de Marco, 2009

Análise sintática (4/5): análise ascendente. YACC. Parser LR(0).

10 - 5a, 02 de Abril, 2009

Análise sintática (5/5): Parser LR(0) - Conjunto de ítens - Análise SLR(1).

11 - 3a, 07 de Abril, 2009

Conclusão sobre Parsing + Análise semântica (1): noções gerais.

12 - 5a, 09 de Abril, 2009

Laboratório - encontro etapa 3 (yacc).

13 - 3a, 14 de Abril, 2009

Análise semântica (2): Implementação de esquemas S e L atribuídos.

14 - 5a, 16 de Abril, 2009

Geração de código (1/6) - código TAC, caso das expressões.

15 - 5a, 23 de Abril, 2009

Geração de código (2/6) - Declarações e Tabelas de símbolo.

16 - 3a, 28 de Abril, 2009

Geração de código (3/6) - Arrays e arrays multidimensionais (3/5). Atributos no Yacc.

17 - 5a, 30 de Abril, 2009

Laboratório – encontro etapa 4 (declaração de variáveis + expressões).

18 - 3a, 05 de Maio, 2009

Geração de código (4/6) - Checagem de tipo.

19 - 5a, 07 de Maio, 2009

Geração de código (5/6) - Expressões booleanas

20 - 3a, 12 de Maio, 2009

Geração de código (6/6) - controle de fluxo.

21 - 5a, 14 de Maio, 2009

Laboratório – encontro etapa 5 (arrays).

22 - 3a, 02 de Junho, 2009

Prova teórica.

23 - 5a, 04 de Junho, 2009

Suporte do ambiente ao runtime (1/2).

24 - 3a, 09 de Junho, 2009

suporte do ambiente ao runtime (2/2) - Geração de código assembly.

25 - 3a, 16 de Junho, 2009

Otimização de código (1).

26 - 5a, 18 de Junho, 2009

Otimização de código (2).

27 - 3a, 23 de Junho, 2009

Laboratório – encontro etapa 6 (controle de fluxo).

28 - 5a, 25 de Junho, 2009

Conclusão: Compiladores e programação em 2008.

29 - 3a, 30 de Junho, 2009

Apresentação final dos projetos (assembly)

30 - 5a, 02 de Julho, 2009

Apresentação final dos projetos (assembly)

TÉCNICAS DE ENSINO (EXPERIÊNCIAS DE APRENDIZAGEM)

A disciplina é apresentada em aulas teórico-práticas, em que se combina a apresentação dos conceitos e técnicas com o desenvolvimento de aplicações pelos alunos.

CRITÉRIOS DE AVALIAÇÃO

Provas Escritas

Será realizada 1 prova teórica P, num dia previamente informado (conforme o Programa acima, no dia 2 de Junho), envolvendo todo o conteúdo das aulas anteriores à prova. O formato da prova inclui perguntas e respostas escritas e exercícios com o uso do computador. Essa prova tem peso 1 sobre o total.

Trabalhos Práticos

Os trabalhos deverão ser realizados no horário das aulas e fora delas e entregues na forma eletrônica (upload no Moodle) nas datas indicadas. Haverá encontros com um tutor (professor, ou aluno de pós-graduação em atividade didática) para apresentação de cada um dos trabalhos. A soma T das notas dos trabalhos tem peso 1 sobre o total.

Ressalta-se que qualquer tentativa de copiar partes dos programas a serem entregues, ou de obtê-las através de práticas contrárias às régras do código disciplinar discente, resultará imediatamente na nota zero na disciplina. (Cita-se aqui o artigo 9 do Código Disciplinar Discente (RESOLUÇÃO Nº 07/2004 do CEPE): “Art. 9º – São infrações disciplinares discentes graves: X – apresentar, em nome próprio, trabalho que não seja de sua autoria;”).

Formação do Conceito Final

A média ponderada das provas e trabalhos (P+ T)/2 será convertida em conceito através da tabela abaixo, levando-se também em conta nesse conceito a participação em aula, interesse, assiduidade e outros critérios subjetivos.

Nota	Conceito
>= 9,0	A
>= 7,5 e < 9,0	B
>= 6,0 e < 7,5	C
< 6,0	D

ATIVIDADES DE RECUPERAÇÃO

Recuperação por falta justificada

No caso de falta justificada por motivo de saúde à prova teórica ou a um encontro de apresentação dos trabalhos, o aluno poderá recuperá-la em data, horário e local a serem marcados pelo professor.

Recuperação do Conceito D

O aluno que obtiver conceito final D poderá recuperá-lo, realizando uma prova de recuperação versando sobre todo o conteúdo do programa. Se a nota obtida nessa prova for igual ou superior a 6,0 o conceito mudará para C.

BIBLIOGRAFIA BÁSICA

1. Aho, Sethi and Ullman ; “Compiladores : princípios, técnicas e ferramentas”, Ed. Guanabara, 1995.
2. Mason, T. e Brown, D., “Lex & Yacc”, O’Reilly, 1991.
3. Grune, D., Bal. H. e Langendoen, K. ; “Projeto moderno de compiladores - Implementação e Aplicações”, Editora Campus, 2001.

Adicionalmente será disponibilizado material suplementar no sistema Moodle de apoio ao ensino (http ://moodle.inf.ufrgs.br), disciplina “Compiladores”.