

PLANO DE ENSINO

INF01033

NOME DA DISCIPLINA: COMPILADORES B

Semestre: 2009/2

Carga horária: 60 h

Créditos: 04

Professor(es): Marcelo Johann

Súmula:

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

Conteúdo Programático:

1. Apresentação da disciplina. Motivação e Introdução a Compiladores.
2. Análise léxica - expressões regulares e autômatos como reconhecedores.
3. Análise sintática - gramáticas livres de contexto, reconhecedores : analisadores *top-down* e *bottom-up* ; conjuntos *First* e *Follow*; tabelas preditivas e LR.
4. Uso das ferramentas *lex* e *yacc* para geração de reconhecedores.
5. Análise semântica ; atributos semânticos herdados e sintetizados ; esquemas S e L atribuídos.
6. Geração de código intermediário para atribuição, checagem de tipos, vetores e estruturas de controle de fluxo de execução (laços, testes).
7. Suporte ao *run-time* - registro de ativação - transferência de parâmetros.
8. Geração de código *assembly* .
9. Otimização de código - bloco básico - laços naturais.
10. Recuperação de erros

Procedimento Didático:

Haverá aulas expositivas e em laboratório 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.

Método de Avaliação:

Provas Escritas

Será realizada 1 prova teórica P, num dia previamente informado, 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, monitor, ou aluno de pós-graduação em atividade didática) para apresentação de cada uma das etapas intermediárias 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 regras 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;”).

PLANO DE ENSINO

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
$\geq 9,0$	A
$\geq 7,5$ e $< 9,0$	B
$\geq 6,0$ e $< 7,5$	C
$< 6,0$	D

ATIVIDADES DE RECUPERAÇÃO

A prova teórica P poderá ser recuperada numa data única, combinada com o professor no fim do semestre. APENAS no caso de o aluno ter obtido pelo menos a nota 6 nos trabalhos ($T \geq 6$).

Cada nota parcial compondo a nota T poderá ser recuperada até o valor máximo de 8/10, pela entrega de uma nova versão do trabalho corrigindo os eventuais defeitos, num prazo estipulado pelo professor.

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.

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.
4. Price, Ana M. A., Toscani, Simão Sirineo. “**Implementação de Linguagens de Programação: Compiladores**”. Porto Alegre, Instituto de Informática da UFRGS/Sagra Luzzatto, 2000.

Bibliografia Complementar:

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

Cronograma por aula:

aula	Dia	Conteúdo
01	18/08	Introdução - Porque estudar compiladores
02	20/08	Estrutura; Análise Léxica; ER; Autômatos; Introdução Lex
03	25/08	Tabela de Símbolos, implementação HASH
04	27/08	Gramáticas Livres do Contexto - revisão e manipulações
05	01/09	Análise sintática - algoritmos
06	03/09	Laboratório 1 - avaliação: análise léxica e tabela Hash
07	08/09	Introdução ao Yacc

PLANO DE ENSINO

08	10/09	Análise sintática - algoritmos
09	15/09	Análise sintática - algoritmos
10	17/09	Análise sintática - algoritmos
11	22/09	Laboratório 2 - avaliação: análise sintática e tabela de símbolos
12	24/09	Árvore de Análise e Tradução dirigida por Sintaxe
13	29/09	Verificações Semânticas e tipos de dados
14	01/10	Verificações Semânticas e tipos de dados
15	06/10	Laboratório 3 - avaliação: Geração da Árvore Sintática
16	08/10	Verificações Semânticas e tipos de dados
17	15/10	Prova P1
18	27/10	Laboratório 4 - avaliação: Verificações Semânticas, Tipos
19	29/10	Geração de Código de três endereços
20	03/11	Geração de Código de três endereços
21	05/11	Geração de Código de três endereços
22	10/11	Laboratório 5 - avaliação: Geração de TACs
23	12/11	Geração de Código Assembly
24	17/11	Geração de Código Assembly
25	19/11	Otimização de Código
26	24/11	Laboratório 6 - avaliação: geração de código assembly
27	26/11	Otimização de Código
28	01/12	Conclusão: tópicos avançados e perspectivas
29	08/12	Apresentação final - otimização, recuperação de erros
30	10/12	Apresentação final - otimização, recuperação de erros
-	17/12	Prova de Recuperação