

Plano de Ensino da Disciplina de Compiladores 2013-2 (Turma A)

A disciplina tem como objetivo apresentar as várias etapas que compõem um compilador – análise léxica, sintática e semântica, otimização e geração de código – de forma que o mesmo possa traduzir um texto escrito em uma linguagem fonte para uma linguagem alvo, em geral uma linguagem de máquina. Descreve-se a seguir a metodologia, as experiências de aprendizagem, os critérios de avaliação, as atividades de recuperação, o cronograma e as referências bibliográficas (básicas e complementares) a serem utilizados nesta disciplina.

1 Metodologia

A metodologia de ensino é materializada através de aulas expositivas e interativas. Os alunos poderão assistir as aulas, fazer perguntas e consultar o material disponibilizado pelo professor, além de consultar as referências bibliográficas disponíveis na biblioteca. Haverá uma série de avaliações ao longo do semestre (veja Seção 3) que servem como verificação e acompanhamento do entendimento dos conceitos da disciplina.

O material apresentado em sala de aula será disponibilizado no ambiente Moodle Institucional¹ da UFRGS. Tal ambiente pode ser acessado por todos os alunos matriculados na disciplina, bastando para tal utilizar o seu número de matrícula (Cartão UFRGS) e sua senha pessoal do Portal do Aluno.

Sugere-se ao aluno que utilize e verifique constantemente o material disponível no ambiente Moodle citado, antes e após as aulas. Nele serão disponibilizados exercícios e atividades extraclasse, notas, especificações de trabalho e demais materiais que sejam necessários. Toda comunicação do professor com os alunos será feita através do Moodle.

2 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. Espera-se que os alunos tenham uma participação ativa nas aulas, realizando anotações, procurando compreender os conteúdos e discutí-los com o professor e colegas. Serão disponibilizados exercícios (teóricos e práticos) e fóruns de discussão como atividades complementares. A participação ativa nessas atividades também é importante para a formação do aluno e serão consideradas pelo professor durante o processo de avaliação. Opcionalmente, os alunos podem:

- Programar um analisador lexical, por exemplo com a ferramenta tradicional Lex, ou pela implementação de qualquer outro algoritmo que possa ser utilizado como analisador léxico
- Programar um analisador sintático, por exemplo com a ferramenta tradicional Yacc, ou pela implementação de qualquer outro algoritmo que possa ser utilizado como um analisador sintático
- Programar ações semânticas para gerar código, simplificado ou não
- Implementar um conversor código abstrato – assembly

3 Critérios de Avaliação

A avaliação da disciplina de compiladores compreende uma prova escrita e um trabalho prático (projeto de compilador).

Será realizada uma prova teórica *PA*, 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. Não será aceito aluno na prova a partir de 15 minutos após o início da mesma.

¹<http://moodleinstitucional.ufrgs.br>

Os trabalhos práticos deverão ser realizados no horário das aulas e fora delas e entregues na forma eletrônica através do Moodle institucional nas datas indicadas. Haverá encontros com um tutor (professor, monitor, ou aluno de pós-graduação em atividade didática) para apresentação de cada um dos trabalhos.

Os seguintes pesos dos critérios de avaliação serão utilizados nesta disciplina:

- Prova (*PA*) correspondendo a 50% da nota final
- Série de trabalhos práticos (*TP*) correspondendo a 50% da nota final

As avaliações parciais *PA*, *TP* receberão uma nota no intervalo [0.0, 10.0].

Média Final

A média final (*MF*), por sua vez, será calculada através da seguinte equação:

$$MF = (PA \times 0.50) + (TP \times 0.50) \quad (1)$$

onde *TP* corresponde a série de trabalhos práticos.

Transformação para Conceitos

A conversão da média final, obtida através da Equação 1, para conceitos é feita por meio da seguinte tabela:

Média Final	Conceito
$9.0 \geq MF \leq 10.0$	A (aprovado)
$7.5 \geq MF < 9.0$	B (aprovado)
$6.0 \geq MF < 7.5$	C (aprovado)
$0.0 \geq MF < 6.0$	Recuperação (veja Seção 4)

Considerações

As notas das provas e trabalhos serão disponibilizadas em até duas semanas depois de sua realização. Pedidos de revisão de notas só serão considerados se o aluno obtiver pelo menos 75% de presenças no momento do pedido. Somente serão calculadas as médias finais daqueles alunos que tiverem, ao longo do semestre, obtido um índice de frequência às aulas igual ou superior a 75% das aulas previstas. Aos que não satisfizerem esse requisito, será atribuído o conceito FF (Falta de Frequência).

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;”

4 Atividades de Recuperação

Os alunos que obtiverem uma média final (*MF*) menor do que 6.0 podem realizar uma única prova de recuperação (**PR**). Note que o aluno só pode realizar a prova de recuperação caso o mesmo tenha obtido pelo menos uma nota *TP* ≥ 6 nos trabalhos práticos. Esta prova avalia toda a disciplina e sua data está prevista no cronograma detalhado na Seção 5. Se a nota obtida nesta prova **PR** for igual ou superior a 6, o conceito do aluno mudará para C.

Recuperação dos Critérios de Avaliação

Não está previsto nenhum tipo de recuperação excepcional para a prova (*PA*) ou para os trabalhos práticos (*TP*), exceto para os casos previstos na legislação:

“A recuperação das provas ou atividades individuais será realizada somente para os casos previstos na legislação: saúde, parto, serviço militar, convocação judicial, luto, etc., devidamente comprovados, através de processo aberto na Junta Médica da UFRGS ou no órgão competente, conforme o caso. Tendo o direito a recuperação, o professor estipulará a data, horário e local de sua realização.”

5 Cronograma e Carga Horária

O calendário letivo 2013/2 estabelece que as aulas irão do dia 5 de agosto até o dia 20 de dezembro. A carga horária desta disciplina é de 60h. A tabela a seguir detalha, dia por dia, o planejamento de todos os encontros.

	Data	Descricao	TP
01	05-08-2013	Aula inaugural: definições, requisitos e posicionamento da disciplina.	0
02	07-08-2013	Análise Léxica 1/2: Introdução, Expressões Regulares, Autômatos Finitos	0
03	12-08-2013	Análise Léxica 2/2: Autômato Finito Determinístico, Flex	1
04	14-08-2013	Análise Sintática 1/5: Introdução, Gramáticas Livres de Contexto, Transformações	1
05	19-08-2013	Apresentação das Etapas 0 e 1 do Projeto de Compilador	2
06	21-08-2013	Análise Sintática 2/5: Análise Descendente Recursiva com Retrocesso	2
07	26-08-2013	Análise Sintática 3/5: análise top-down - definição de First/Follow.	2
08	28-08-2013	Análise Sintática 4/5: análise top-down com tabela preditiva.	2
09	02-09-2013	Apresentação da Etapa 2 do Projeto de Compilador	3
10	04-09-2013	Árvore Sintática Abstrata (AST)	3
11	09-09-2013	Análise Sintática 5/5: análise ascendente.	3
12	11-09-2013	Análise Sintática 6/5: Parser LR(0) - Conjunto de itens - Análise SLR(1).	3
13	16-09-2013	Fim parsing + Exercícios	3
14	18-09-2013	Apresentação da Etapa 3 do Projeto de Compilador	4
15	23-09-2013	Análise semântica 1/2 - conceitos gerais.	4
16	25-09-2013	Análise semântica 2/2 - conceitos gerais.	4
17	30-09-2013	Geração de código 1/6 - código TAC, caso das expressões.	4
18	02-10-2013	Geração de código 2/6 - declarações, tabelas de símbolos - atribuição.	4
19	07-10-2013	Apresentação da Etapa 4 do Projeto de Compilador	5
20	09-10-2013	Geração de código 3/6 - Arrays multidimensionais.	5
21	14-10-2013	Geração de código 4/6 - Checagem de tipo.	5
22	16-10-2013	Geração de código 5/6 - Expressões booleanas	5
23	30-10-2013	Suporte do ambiente ao runtime (1/2).	5

24	04-11-2013	Suporte do ambiente ao runtime (2/2) - Geração de código assembly.	5
25	06-11-2013	Apresentação da Etapa 5 do Projeto de Compilador	6
26	11-11-2013	Revisão Geral	6
27	13-11-2013	Prova teórica PA	6
28	18-11-2013	Otimização de código (1).	6
29	20-11-2013	Otimização de código (2).	6
30	25-11-2013	Apresentação da Etapa 6 do Projeto de Compilador	7
31	27-11-2013	Otimização de código (3).	7
32	02-12-2013	Conclusão da disciplina de compiladores: discussão	7
33	04-12-2013	Apresentação Final do Projeto de Compilador	
34	09-12-2013	Prova de Recuperação – PR	
35	11-12-2013		
36	16-12-2013		
37	18-12-2013		

6 Referências

As referências básicas essenciais são as seguintes:

- Aho and Sethi and Ullman. *Compiladores - Princípios, Técnicas e Ferramentas*. Addison-Wesley, 2008
- Grune and Dick. *Projeto moderno de compiladores – Implementação e Aplicações*. Campus, 2001
- John R. Levine and Tony Mason and Doug Brown. *Lex & Yacc*. O'Reilly, 1992

A referência essencial é a seguinte:

- Keith Cooper and Linda Torczon. *Engineering a Compiler*. 2 edition, 2011

A referência complementar é a seguinte:

- Kernighan, Brian W. and Ritchie, Dennis M. *The C programming language*. Prentice Hall, 1989