



UNIVERSIDADE DE SÃO PAULO - ICMC

Departamento de Ciências de Computação

SCC 206 – Introdução a Compilação - 1º Sem /2010

PROFa: Sandra Aluisio

WIKI: <http://wiki.icmc.usp.br/index.php/SCC-206>

I – PROGRAMA

- **Conceitos básicos:** compiladores, interpretadores, montadores, filtros, pré-processadores; estrutura conceitual de um compilador; formas de organização de um compilador; processo de execução de uma linguagem de alto-nível; compilador cruzado (cross-compiler), auto-compilável (bootstrapping), auto-residente e compiler compilers (ou translator writing systems, ou compiler generators).

- **Definição de uma linguagem** – Pascal Simplificado (PS) – e suas extensões para cada aluno

Relatório do Projeto: 1. Especificações Gerais do Compilador; 2. Especificações da Linguagem Fonte com as extensões do grupo; 3. Especificações da Linguagem Objeto (P1)

- **Análise Léxica:** vantagens da separação entre as análises léxica e sintática; erros Léxicos; especificação e reconhecimento de tokens; formas de implementar os autômatos finitos que reconhecem os tokens do programa-fonte.

- **Análise Sintática Descendente** (ASD ou *TOP-DOWN*)

- com Retrocesso

- sem Retrocesso (*Predictive Parsers*):

- Procedimentos Recursivos

- Dirigida por Tabela (LL(k))

Recuperação de Erros em ASD com Procedimentos Recursivos

- **Introdução ao compiler compiler JavaCC** (<https://javacc.dev.java.net/>)

Especificação da Análise Léxica e Sintática com as extensões do grupo (P2)

- **Introdução a Análise Ascendente** (ASA ou *BOTTOM-UP*)

- **Análise Semântica, Tratamento de Erros Dependentes de Contexto e Checagem de Tipos:** o componente semântico de uma LP; Tarefas da Análise Semântica; Técnicas de Implementação para Tabelas de Símbolos; Ações Semânticas em Compiladores Dirigidos por Sintaxe e Erros da Análise Semântica; Tarefas da Checagem de Tipos; Checagem de tipos Estática versus Dinâmica; Tipo de uma variável; Compatibilidade de tipos; conversões de tipos; Operadores sobrecarregados.

Projeto de um Analisador Semântico para o PS com extensões e de Rotinas de Checagem Estática de Tipos (P3)

- Ambientes de Execução e Geração de Código para uma máquina Hipotética (MEPA): rotinas de suporte ao tempo de execução em várias linguagens (Fortran, C, Pascal); *bindings*; memória em tempo de execução; apresentação da MEPA e suas instruções.

Projeto de um Gerador de Código para a máquina MEPA (P4)

II – BIBLIOGRAFIA

- Aho, A. V. et alli - Compilers: Principles, Techniques and Tools. Addison-Wesley Publishing Company, 1986. (Dragãozinho Vermelho em INGLÊS)
- Aho, Lam, Sethi & Ullman. Compilers: Principles, Techniques, and Tools; 2nd Edition, Addison-Wesley Publishing Company, 2006. (Dragãozinho Roxo em INGLÊS)
- <http://dragonbook.stanford.edu/>
- Aho, A. V. et alli - Compiladores - Princípios, Técnicas e Ferramentas, 2ª Edição, Ed. Pearson, 2007. (Dragãozinho Roxo em PORTUGUÊS)
- Aho, A.V.; Ullman, J.D.; Sethi, R. (1995). Compiladores: Princípios, Técnicas e Ferramentas. Editora LTC. (Dragãozinho Vermelho em PORTUGUÊS)
- Kowaltowsky, T. - Implementação de Linguagens de Programação, Guanabara Dois, 1983.
- Louden, K.C. (2004). Compiladores: Princípios e Práticas. Editora Thomson Learning.
- Price, A.M.A. e Toscani, S.S. (2001). Implementação de Linguagens de Programação: Compilador. Editora Sagra Luzzatto.

III - CRITÉRIO DE AVALIAÇÃO

PROVAS: Haverá 2 provas cujas datas de realização estão marcadas a seguir. Cada prova vale de 0 a 10. Não há substitutiva.

1ª Prova: 4/5 das 17:20 – 19:00 2ª Prova: 29/6

PROJETO: Um compilador que traduz programas da linguagem Pascal Simplificado (PS) em código de um compilador hipotético à pilha chamado MEPA (Máquina de Execução para Pascal). Para a implementação desse projeto será fornecida a sintaxe da linguagem fonte – PS – em notação EBNF. As características gerais da MEPA e a interpretação das instruções serão vistas durante o curso. As instruções da MEPA encontram-se em Kowaltowsky, T. - Implementação de Linguagens de Programação, Guanabara Dois, 1983.

O projeto será desenvolvido por uma equipe de 3 alunos (se sobraem alunos faremos grupos de 4 ou de 2). O projeto é dividido em 4 entregas, definidas no cronograma abaixo. Cada uma vale de 0 a 10 e suas correções serão baseadas em critérios explícitos definidos previamente, juntamente com a especificação da entrega.

LISTAS DE EXERCÍCIOS. Haverá várias listas. Espera-se que os alunos resolvam as listas e tirem as dúvidas com o monitor PAE ou professor.

CÁLCULO DA MÉDIA:

MP = Média Aritmética das Provas

MT = Média Ponderada das Notas dos Projetos: 5%, 15%, 40%, 40%

MF = Média Final

Se $MP \geq 5$ e $MT \geq 5$

$$MF = (6MP + 4MT)/10$$

Se $MP < 5$ ou $MT < 5$

MF = menor valor entre MP e MT

RECUPERAÇÃO: Só terão direito à recuperação os alunos com $3.0 \leq MF < 5.0$ e frequência $\geq 70\%$. Média = $MF + (Rec/2,5)$, se $Rec \geq 7,5$; ou Média = $\text{Max}\{MF, Rec\}$, se $Rec < 5,0$; ou Média = $5,0$, se $5,0 \leq Rec < 7,5$.

Cronograma

Data	Aula	Tópico	Trabalho	Data Entrega
23/2	1	Apresentação do Curso, da linguagem para o projeto, Conceitos básicos		
2/3	2	Especificações Análise léxica	P1	
9/3	3	AS Descendente		P1
16/3	4	AS Descendente		
23/3	5	Javacc	P2	
30/3	Não há AULA	Semana Santa		
6/4	Não há AULA	Conferência		
13/4	6	AS Ascendente		
20/4	7	AS Ascendente		P2
27/4	Não há AULA	Conferência		
4/5	8	ASem PROVA 1 – das 17:20 – 19:00 h		
11/5	9	ASem		
18/5	Não há AULA	Conferência		
25/5	10	ASem	P3	
1/6	11	Checagem de Tipos		
8/6	12	Ambientes de Execução & Geração Código		P3
15/6	13	Ambientes de Execução & Geração Código	P4	
22/6	14	Ambientes de Execução & Geração Código		
29/6	15	PROVA 2 – Das 16:20 – 19:00 h		
6/7	16			P4