

# MO615/MC900 Implementação de Linguagens II (2s12)

## Notícias

- **30/07 Seja bem-vindo!**
- **30/07 Esta disciplina não tem MO403 Implementação de Linguagens I como pré-requisito. Apenas um curso básico de compiladores na graduação é necessário para cursá-la.**
- **29/08 Aula de amanhã Qui. (30/8) cancelada**

## Ementa

Esta disciplina cobrirá técnicas, métodos e algoritmos utilizados para otimizar o desempenho de programas. Serão cobertas técnicas de análise de fluxo de dados e otimização independente de máquina (ex. eliminação de expressões comuns), alocação de registradores (via coloração de grafos), otimização de laços (ex. remoção de invariantes), gerenciamento de memória (ex. coleta de lixo) e compilação orientada a objetos. Estudaremos também algoritmos para detecção e extração de paralelismo em arquiteturas multicore (ex. software pipelining), MIMD (ex. list scheduling) e GPUs (ex. strip mining).

## Horários

- Aulas: Ter. e Qui.
- Sala: 316 (IC3.5)
- Horários: 8:00 - 10:00
- Instrutor: Prof. Guido Araújo
- Atendimento: agende via e-mail.

## Monitor

- Marcio Pereira
- Atendimento: Ter. e Qui. 13:00 - 14:00, sala 60

## Bibliografia

O curso será fortemente baseado nos livros abaixo.

- Sethi and Ullman. Compilers: Principles, techniques and tools.. Addison-Wesley.

- Steven S. Muchnick. Advanced Compiler Design and Implementation. Morgan Kaufmann
- Michael Wolfe. High Performance Compilers for Parallel Computing.
- Andrew Appel. Modern Compiler Implementation in Java. Second Edition Cambridge

## Slides

As aulas utilizarão um conjunto de slides, disponíveis abaixo nesta página. Se necessário, notas de aula adicionais, bem como artigos discutidos em classe serão disponibilizados.

## Vídeos

Algumas aulas desta disciplinas estão disponíveis em vídeo. Salientamos que estes vídeos somente poderão ser acessados por alunos regularmente matriculados na disciplina, e que o material é propriedade intelectual do professor e da UNICAMP, não podendo ser distribuído sem a autorização destes.

## Newsgroup

O newsgroup desta disciplina pode ser encontrado [aqui](#).

## Exames

A avaliação será baseada em dois exames e em um seminário com os seguintes pesos: Exame I (35%), Exame II (35%) e Seminário (30%). Fraudes nos exames não serão aceitos e será atribuída a média zero a todos os envolvidos. Pedidos de revisão de prova devem ser feitos através de e-mail (assunto MO615), até 48 horas após a divulgação do resultado do exame. O conceito final será atribuído de acordo com as faixas abaixo, considerando-se a média final (M), normalizada em relação à turma:

- A:  $M > 8.5$
- B:  $7 < M \leq 8.5$
- C:  $5 \leq M \leq 7.0$
- D:  $M < 5.0$

As notas estarão disponíveis nesta Tabela, duas semanas após a realização do Exame.

## Exercícios

Consideramos os exercícios do livro-texto como fundamentais para o bom aprendizado da matéria e a realização dos exames. Recomenda-se fortemente que sejam resolvidos os problemas sugeridos aqui nesta seção.

- Aho (2nd Edition): 9.2.1,9.2.2,9.2.3
- Appel (2nd Edition) : 9.1, 9.3, 10.1,10.5, 11.1, 11.2, 11.3,11.4, 17.1, 17.2, 17.5

## Cronograma

- 31/7 Introdução, IR, Árvores e DAGs,
- 2/8 Blocos básicos e CFG
- 7/8 Conceitos básicos de otimização de código
- 9/8 Introdução à análise de fluxo de dados (DFA) e definições alcançáveis e DU-chains
- 14/8 Análise expressões disponíveis
- 16/8 Análise de variáveis vivias e DU-chains
- 21/8 Eliminação de expressões comuns, código morto e constantes
- 23/8 Alocação global de registradores
- 28/8 Alocação global de registradores (cont.)
- 30/8 Laços no CFG, dominadores e árvore de dominadores
- 4/9 Laços naturais, pre-headers, CFG redutíveis e laços aninhados
- 6/9 Invariantes de laço, detecção de variáveis de indução
- 11/9 Strength reduction, eliminação de variáveis de indução e code motion
- 13/9 Exercícios
- 18/9 Exame I
- 20/9 Transformação de laços: loop unrolling, re-ordenação de sentenças, loop unswitching
- 25/9 Transformação de laços: loop peeling, loop splitting, loop fusion, loop interchange, strip-mining e loop blocking
- 27/9 Seleção de instruções

- 2/10 Escalonamento de instruções: dependência de dados e escalonamento de desvios
- 4/10 Escalonamento de instruções: list e trace scheduling
- 9/10 Escalonamento de instruções: software pipelining
- 11/10 Paralelismo: Laços DOALL, DOACROSS e DOPIPE
- 16/10 Forma SSA
- 18/10 Otimizações em SSA
- 23/10 Otimizações em SSA
- 25/10 Exercícios
- 30/10 Alias Analysis
- 1/11 Coleta de lixo
- 6/11 Coleta de lixo (cont.)
- 8/11 Implementação de mecanismos de orientação a objetos
- 13/11 Seminário I
- 15/11 Seminário II
- 20/11 Feriado
- 22/11 Exame II
- 27/11 Divulgação da média