

# MATA61 - Compiladores @ UFBA

## Objetivos

- **Parte teórica:** Entender os conceitos fundamentais envolvidos na implementação de linguagens de programação;
- **Parte prática:** Implementar um compilador aplicando as técnicas de análise e síntese/geração de código.

## Projeto Prático

Projetar e implementar um compilador (incluindo Analisador Léxico, Analisador Sintático/Semântico, e Gerador de Código) para uma linguagem simples procedimental. Detalhes do projeto [aqui](#).

## Provas/trabalhos

- **Trabalho 1:** Analisador Léxico
- **Prova 1:** Análise Léxica, Análise Sintática (Top-down e Bottom-up), Análise Semântica (Escopo e Tipos)
- **Trabalho 2:** Analisador Sintático/Semântico
- **Prova 2:** Ambiente de execução, Geração de Código, Otimização de código, Alocação de registradores, Gerenciamento Automático de Memória
- **Trabalho 3 (final):** Gerador de Código

Como regra geral, não serão avaliados trabalhos entregues fora do prazo. Qualquer dificuldade ou dúvida, entre em contato com o professor.

## Avaliação

- 10% Listas de Exercícios (*individual*)
- 25% Primeira prova (*individual*)
- 25% Segunda prova (*individual*)
- 40% Trabalhos práticos (*individual ou em dupla*)

## Conteúdo da Disciplina (Notas de aula)

## Conteúdo da Disciplina (Notas de aula)

### ≡ MATA61 Compiladores

#### 1. Visão geral da disciplina e conceitos básicos

1. Introdução a Tradução de Programas
2. Papel do Compilador/Interpretador, Conceitos Básicos
3. Estrutura e fases de um compilador

#### 2. Análise Léxica

1. Conceitos de análise léxica
2. Exemplos envolvendo análise léxica
3. Linguagens regulares
4. Exemplos de especificação léxica
5. Especificação de lexemas/tokens em linguagem de programação
6. Automato Finito
7. Mapeando expressões regulares em automatos finitos
8. Implementação de automatos finitos
9. Exemplos de códigos de implementação

#### 3. Análise sintática: Introdução

1. Conceitos gerais
2. Gramáticas livres de contexto
3. Derivações
4. Ambiguidade
5. Representação abstrata/compacta da árvore sintática

#### 4. Análise Sintática: Top-Down

1. Algoritmo recursivo descendente
2. Limitação do parser recursivo descendente
3. Problema da recursão à esquerda
4. Parser preditivo LL(1)
5. Conjunto First
6. Conjunto Follow
7. Construção tabela do parser LL(1)

#### 5. Análise Sintática: Bottom-Up

1. Estratégia de Parser Shift-reduce
2. Conceito de Handle

### ≡ 3. Reconhecimento de um Handle

#### 4. Reconhecendo Prefixos Viáveis

#### 5. Parser SLR ("Simple LR")

#### 6. Exemplo de funcionamento do parser SLR

#### 7. Melhorias para o parser SLR

#### 8. LR(1), LALR(1) parsers

#### 9. Geradores de analisadores sintáticos

### 6. Análise semântica

#### 1. Escopo e Tabela de símbolos

#### 2. Checagem de tipos

#### 3. Ambientes de tipos

#### 4. Sub-tipos

#### 5. Implementação de checagem de tipos

### 7. Ambiente de execução

#### 1. Ativações

#### 2. Registros de Ativação

#### 3. Variáveis Globais e Heap

### 8. Geração de código

#### 1. Máquinas de Pilha

#### 2. Linguagem MIPS assembly

#### 3. Implementação usando Máquina de Pilha

#### 4. Modelos de Geração de Código

### 9. Otimização de Código: Local

#### 1. Código Intermediário

#### 2. Técnicas de otimização local

#### 3. Otimização "peephole"

### 10. Otimização de Código: Global

#### 1. Análise Data flow

#### 2. Propagação global de constantes

#### 3. Análise Liveness

### 11. Alocação de Registradores

#### 1. Hierarquia de memória

- 1. Técnica de Gerenciamento de Memória
- 2. Definição do Problema
- 3. Método de Coloração em grafos
- 4. Técnica de Spilling

## 12. Gerenciamento Automático de Memória

- 1. Definição do Problema
- 2. Técnica Mark and Sweep
- 3. Técnica Stop and Copy
- 4. Técnica Reference Counting

## Livro para leitura/consulta (opcional)

Livro do "Dragao": Aho, Lam, Sethi, and Ullman. Compiladores: princípios, técnicas e ferramentas. 2a edição, Addison-Wesley, 2008.

- Analise Lexica e Automato Finito
  - Seções de Capítulos: 3.1, 3.3, 3.4, 3.6, 3.7, 3.8
- Analise Sintatica
  - Seções de Capítulos: 4.1-4.6, 4.8.1, 4.8.2
- Analise Semantica e Tipos
  - Seções de Capítulos: 5.1-5.3 6.3, 6.5
- Ambiente de Execucao e Geracao deCodigo
  - Seções de Capítulos: 6.2, 7.1-7.4, 8.1-8.3, 8.6
- Otimizacao de codigo
  - Seções de Capítulos: 8.5, 8.7, 9.1-9.4
- Alocação de registradores e Gerenciamento automatico de memoria
  - Seções de Capítulos: 8.8, e 7.5-7.7