



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE RORAIMA
CENTRO DE CIÊNCIA E TECNOLOGIA
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO



Disciplina Construção de Compiladores
Professor Luciano Ferreira Silva

Trabalho Final – **Valor: 10,0 pontos** – **Nota:** _____

Aluno: _____ **Número:** _____ **Data:** __/__/__

A Prova – Construir, documentar e defender o sistema *CompilerExpressions*.

O *CompilerExpressions* será um compilador que visa traduzir um código-fonte, contendo expressões matemáticas e outros comandos que permitam trabalhar com as mesmas, para um código-objeto em linguagem simbólica.

A construção

Ao desenvolver o *CompilerExpressions* o aluno deverá atentar-se para as seguintes especificações:

1. O *CompilerExpressions* deve apresentar uma interface gráfica que permita ao seu usuário utilizar todas as suas funcionalidades.
2. O *CompilerExpressions* deve permitir ao seu usuário programar:
 - ✓ a partir em um arquivo .txt, e posteriormente carregar o seu código para compilação;
 - ✓ ou em uma interface de editoração própria, que possibilite o envio do código para compilação ao seu término.
 - ✓ É vetada a opção de o usuário inserir o seu código-fonte direto no código-fonte do *CompilerExpressions*.
3. Após a captação do código-fonte do usuário o *CompilerExpressions* deve, obrigatoriamente, realizar as análises léxica, sintática e semântica, a geração de código intermediário, a otimização deste código, e a produção do código em linguagem simbólica. Para tanto o aluno deve considerar as seguintes premissas:
 - i. Na análise léxica, após a separação de lexemas, os tokens devem ser classificados de acordo com gramáticas regulares criadas pelo próprio aluno. É obrigatório o uso de pelo menos uma gramática regular na análise léxica;
 - ii. Na análise sintática o aluno deverá optar pelo uso de analisadores sintáticos preditivos ou de precedência fraca, ou pelo uso de ambos, caso ache necessário. A sintaxe da linguagem fonte, regida por gramáticas livres de contexto, também é de responsabilidade do aluno. É obrigatório o uso de pelo menos uma gramática livre de contexto na análise sintática.
 - iii. O aluno deverá implementar no mínimo um tópico da análise semântica.
 - iv. O aluno deverá gerar códigos inicialmente no formato intermediário, podendo ser na forma de arquivos texto ou tabelas quádruplas ou triplas.
 - v. O aluno deve implementar no mínimo um tópico da otimização independente de máquina no código intermediário.
 - vi. Por fim o aluno deve escolher uma arquitetura de um processador e gerar o código em linguagem simbólica obedecendo aos seus padrões. Este código deverá ser gravado em um arquivo .txt e/ou apresentado na interface.

4. O programa deverá emitir mensagens corretivas que permitam aos usuários detectar erros léxicos, sintáticos e semânticos.

Referente a esta fase o aluno deverá entregar o *CompilerExpressions* funcionando, submetendo para análise o seu executável e código-fonte. A qualidade do programa e suas funcionalidades serão pesos importantes na avaliação.

A documentação

O aluno deverá documentar todas as etapas da construção do *CompilerExpressions*, em atentando-se para no mínimo:

1. Explicar o funcionamento e os comandos da linguagem fonte;
2. Apresentar as gramáticas regulares que regem o analisador léxico, assim como explicar o seu funcionamento;
3. Apresentar as gramáticas livres de contexto que regem o analisador sintático, assim como explicar o seu funcionamento;
4. Explicar os tópicos utilizados da análise semântica;
5. Apresentar os detalhes da geração de código intermediário;
6. Explicar os tópicos utilizados da otimização independente de máquina;
7. Apresentar os detalhes da geração de código na linguagem simbólica, a explicitando inclusive as especificados da arquitetura do processador escolhido;
8. Explicar e apresentar as mensagens corretivas emitidas pelo *CompilerExpressions* para o usuário.

Nesta fase o aluno deverá entregar um relatório impresso e digital. Sendo que, a sua completude, detalhes e organização serão quesitos fundamentais na avaliação.

A defesa

No momento da entrega do *CompilerExpressions* e o relatório o aluno será argüido, de maneira individual, executando o sistema desenvolvido e respondendo perguntas a seu respeito. Serão avaliados o conhecimento do conteúdo, a clareza, objetividade e segurança nas suas respostas.

Observação

As possibilidades de construir compiladores diferentes são enormes, portanto, o façam.