

Projeto de Compilador

Entrega, Avaliação, Regras

Prof. Lucas Mello Schnorr
schnorr@inf.ufrgs.br

1 Introdução

O trabalho consiste no projeto e implementação de um compilador para uma determinada gramática de linguagem de programação especificada em cada semestre. Este documento estabelece as regras gerais do projeto de compilador, com links para especificação das etapas. A entrega é realizada através de link específico no Moodle, pela submissão de um arquivo em formato `tgz` conforme detalhes abaixo. O projeto compreende seis etapas (conforme tabela abaixo), sendo que o peso de cada etapa é sempre o mesmo.

Etapa	Descrição e Especificação	Peso
E1	Análise Léxica	10
E2	Análise Sintática	10
E3	Árvore Sintática Abstrata	10
E4	Análise Semântica	10
E5	Geração de Código	10
E6	Geração de Assembly	10

A nota máxima do projeto de compilador é 60. Esta será utilizada para integralizar a nota final da disciplina conforme o plano.

2 Procedimento de Entrega

O seu diretório de trabalho deve conter um arquivo `Makefile` com todos os comandos necessários à compilação do seu programa, automatizados através da chamada `make`, incluindo a invocação de `flex`, `bison` e `gcc`. Recomenda-se fortemente a construção de um arquivo `Makefile` que siga a lógica da compilação parcial de um projeto. Sugere-se a leitura do *A Simple Makefile Tutorial* para aqueles que já tem ou não experiência com esta ferramenta.

A execução de `make` deve gerar um executável no mesmo diretório de trabalho chamado exatamente `etapaX` (onde `X` é o dígito identificador da etapa). Exemplo: para a E4, o arquivo binário a ser executado após compilação deve ser `etapa4`.

Cada grupo deve compactar o conteúdo do seu diretório de trabalho executando o comando `tar cvzf etapaX.tgz` . (onde `X` é o dígito identificador da etapa) **dentro do mesmo diretório**. Não utilizem outros programas, formatos, comandos, nomes ou organizações de diretórios para esse operação. Será gerado um arquivo chamado `etapaX.tgz`. O arquivo `etapaX.tgz` deve ser enviado pelo Moodle (no link referente à etapa), **com tamanho máximo de 100KBytes**. Ao submeter solução para qualquer tarefa desta disciplina, tu declaras: (1) que a submissão reflete a solução de sua própria autoria para o problema proposto; (2) que a mesma não foi copiada ou extraída, seja parcial ou integralmente, de nenhuma fonte além daquelas públicas consultadas e corretamente referenciadas no trabalho; e (3) estar ciente de que o plágio é uma infração grave segundo o Código Disciplinar Discente da UFRGS. Caso concordes com esta declaração, utilize o link disponibilizado no moodle para realizar a entrega do arquivo.

3 Formação dos Grupos

Os trabalhos devem ser realizados em grupos de **dois alunos**. Não serão aceitos grupos de três alunos ou trabalhos individuais, exceto para acomodar número ímpar de alunos na turma. A definição de grupos é definitiva, feita pelo moodle em link apropriado, válida do início ao fim do semestre.

4 Regras de desenvolvimento

Existe uma flexibilidade na organização do código fonte, mas temos regras a serem observadas para permitir que testes automáticos possam ser realizados corretamente.

Remova todos os comandos `printf` que lá estão por quaisquer questões, antes da submissão, pois o compilador deve ser silencioso salvo quando indicado o contrário na especificação individual de cada etapa. Utilize somente as ferramentas padrão, como `make`, `flex`, `bison` e `gcc`. Os arquivos fornecidos em cada uma das especificações (por exemplo, `tokens.h` na E1) não devem ser modificados. Não use nenhuma estrutura hierárquica de diretórios. Todos os seus arquivos, incluindo fontes, teste, `Makefile`, executável, devem estar no mesmo diretório de trabalho. Eles serão compactados pelo seu comando `tar` e descompactados por um comando correspondente dentro de um único diretório onde serão testados.

5 Avaliação das Etapas

Cada etapa do projeto de compiladores é avaliada objetiva e subjetivamente, com pesos iguais. A **nota objetiva** é definida por testes automáticos (quando possível) que verificam as soluções dos alunos. A **nota subjetiva** é definida pelo professor, mediante revisão do código pelo professor. Esta nota subjetiva compreende a originalidade, a qualidade do código, a legibilidade, os comentários assim como a organização geral e recursos adicionais. Membros do mesmo grupo podem receber notas diferentes.

6 Política de Atraso

Todas as etapas do projeto de compilador tem a mesma política de atraso. As soluções de cada etapa devem ser enviadas até as datas de entrega especificadas neste local. As soluções recebidas antes da data limite serão avaliadas normalmente, até 100%. Caso a nota de uma etapa entregue no prazo e avaliada seja inferior a 6.0, o professor acordará um prazo com os alunos para que eles possam recuperá-la em até 80% do valor integral daquela etapa. Caso o grupo não entregue na data estipulada, é acordado uma extensão automática com prazo adicional de uma semana para fazer uma **entrega em atraso**. Neste caso, o peso após a avaliação é de 80%, sem novos prazos adicionais de recuperação. Qualquer solução recebida após uma semana da data limite inicialmente estabelecida receberá nota zero.