UFRGS - Informática - INF01147 - Compiladores - 2025/1
Prof. Marcelo Johann

Trabalho Prático - Instruções para Entrega da Etapa 1: Análise Léxica e Inicialização de Tabela de Símbolos

Resumo:
Este texto define as regras, formatos e procedimentos necessários para a entrega da primeira etapa do trabalho de desenvolvimento do compilador.

Definições neste texto:
Os tópicos tratados são os seguintes:

a. formato de entrega;
b. prazo;
c. organização do código e compilação;
d. recomendações e código de ética;

Formato de Entrega
Cada aluno deve compactar o conteúdo do seu diretório de trabalho executando o comando "`tar cvzf etapa1.tgz .`" dentro do mesmo diretório. Não utilizem outros programas, formatos, comandos, nomes ou organizações de diretórios para esse operação. Será gerado um arquivo chamado `etapa1.tgz`, o qual deve ser copiado para o servidor *http* através do comando "`sftp username@html.inf.ufrgs.br`", deixando-o acessível dentro de um subdiretório com um **nome-chave (**`key`**)**, dentro do diretório `public_html`. Esse nome-chave é um código alfanumérico individual de cada aluno e será enviado pelo professor por uma mensagem particular. Ou seja, dentro do *sftp*, executem "`cd public_html`", "`mkdir key`", "`cd key`" e finalmente "`put etapa1.tgz`". Vocês devem verificar as permissões de acesso ao arquivo, corrigi-las se necessário com o comando "`chmod 744`", e recomenda-se que testem o acesso à partir de um navegador. O seu arquivo deve ficar acessível com um navegador em um endereço do tipo: <http://www.inf.ufrgs.br/username/key/etapa1.tgz>. É importante também que a pasta com o nome da chave possa ter a permissão de execução (para que o servidor *http* acesse, mas que *não* possa ser lida. Assim os diretórios "`key`" e principalmente "`public_html`", devem ter permissões 711 - somente execução, não leitura.

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` e `g++`. A execução de `make` deve gerar um executável no mesmo diretório de trabalho chamado apenas e exatamente "`etapa1`". O arquivo `Makefile` também deve conter uma definição `clean`, que será usada para apagar os arquivos intermediários e executável e permitir uma compilação nova completa.

Prazos

A etapa 1 do trabalho deve ser colocada no servidor *http* até as 23:59 horas do dia 28/03/2025, de acordo com o cronograma desse semestre, para que o trabalho seja avaliado de forma automática e pelo professor de forma remota.

Organização do Código

Existe uma certa flexibilidade na organização do código fonte, mas há uma série de regras que devem ser observadas para permitir os testes corretos nos *scripts* automáticos.

a. O trabalho deste semestre será desenvolvido em linguagem C++ 2011, com a biblioteca *STL* (*Standard Template Library*);
b. Somente as ferramentas *make*, *flex* e *g++* devem ser utilizadas e o compilador C++ deve ser chamado com `"g++ -std=c++11 -Wall"`;
c. O código será compilado na máquina do professor, usando Mac OSX, que é compatível com Unix Posix, e usará as versões: GNU Make 3.81, flex 2.6.4 Apple (flex-34), Apple clang version 14.0.0, e bison (GNU Bison) 2.3 à partir da etapa 2;
d. código de *tokens.h* não deve ser modificado, caso contrário os valores retornados não poderão ser verificados;
e. Deve haver um arquivo *main.cpp* separado contendo unicamente a função *main* com as suas chamadas de teste para *yylex*, e nenhuma outra funcionalidade do seu analisador deve estar implementada nesse arquivo ou na *main*.
f. O código do seu arquivo *main.cpp* não pode incluir outros fontes seus com a diretiva `#include` para completar a compilação, pois ele será substituído. Você pode fazer o contrário, incluir *main.cpp* no restante do código, ou compilar os fontes separadamente para código objeto (`g++ -c`);
g. Você deve implementar uma função chamada `void initMe(void)` de onde qualquer código de inicialização necessário deve ser chamado. Essa função será chamada pela *main* testadora substituída pelo professor. Você precisa implementar essa função mesmo não dependa dela, pois será chamada pelos testes automatizados.
h. 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.

Recomendações e código de Ética

Executem vários testes. Verifiquem a conformidade com cada uma das regras desse formato e da especificação da etapa. Verifiquem se o trabalho pode ser compilado e rodado em outro sistema fora o usado para desenvolvimento. Verifiquem o acesso ao arquivo disponibilizado no servidor *http*.

Incluam os nomes de vocês como autores em comentários no início de todos os códigos fonte do seu trabalho. Os trabalhos devem ser inteiramente desenvolvidos pelos alunos. A discussão entre alunos da mesma turma ou de outras turmas, para aprendizagem, é

incentivada, assim como consulta a fóruns de discussão e ferramentas de IA. O código entregue, entretanto, deve ser de completa autoria da dupla de alunos que assinam o trabalho. Segundo o Código Disciplinar discente da UFRGS, "Art. 9o - São infrações discentes graves: ... IX – plagiar, total ou parcialmente, obras literárias, artísticas, científicas, técnicas ou culturais; X – apresentar, em nome próprio, trabalho que não seja de sua autoria;..." (<https://www.ufrgs.br/demat/codigo-disciplinar-discente/>). Se excepcionalmente o trabalho contiver algum trecho como mais de uma linha ou uma função que foi copiada de qualquer outra fonte, incluindo material de ensino da disciplina, este trecho deve ser claramente identificado e a fonte referenciada.

Retirem dúvidas com o professor antes do prazo final.

Porto Alegre, 20 de Março de 2025