

EA876 - Trabalho 1

O problema com o qual vamos lidar neste trabalho é a geração de código em linguagem assembly. Vamos usar assembly de ARM, unicamente pelo motivo de existir um simulador livre, multiplataforma e fácil de usar (você provavelmente verá que mudar de arquitetura para AVR ou x86 é muito simples).

O programa a ser gerado deve converter expressões matemáticas para um código ARM equivalente.

Restrições e anotações

1. Este trabalho tem o objetivo exercitar a conversão de código em uma linguagem (expressões matemáticas) para outra (assembly ARM).
2. O código assembly deverá executar no simulador Visual, disponível online em <https://salmanarif.bitbucket.io/visual/>. Trata-se de um subconjunto das instruções ARM.
3. Você não deve “otimizar” seu código simplificando a expressão, e sim gerá-lo de forma a executar todas as operações.
4. O grupo deverá implementar sua solução usando Lex/Flex, Yacc/Bison e C.
5. O resultado final da expressão deve ser guardado no registrador de propósito geral R0.
6. A calculadora deve suportar soma, subtração, parênteses e multiplicação.
7. No lugar da multiplicação (ou, se o grupo quiser, em adição a ela), o grupo pode optar por implementar a divisão inteira, o operador resto de divisão ou outro operador matemático que não corresponda diretamente a uma instrução específica da arquitetura ARM.

O que entregar

1. O código-fonte deverá ser entregue comentado de forma que seja possível entender o raciocínio do grupo através dos comentários.
2. Um arquivo Makefile e instruções de uso para permitir testar o programa enviado.
3. O grupo deverá entregar um relatório curto (de até 1 página, em formato coluna dupla), em formato **PDF**.
4. O código e o relatório deverão ser entregues num arquivo .zip através do sistema do Classroom. Por favor, deixe o relatório num diretório chamado **doc** os códigos fontes num diretório chamado **src** e o Makefile no diretório raiz.

Sobre o relatório

O relatório é um texto *científico* e *objetivo* e *auto-contido* que tem o propósito de mostrar que a solução adotada pelo grupo é relevante e correta. Ele deve ser dividido em três partes:

1. Introdução: como o problema foi modelado computacionalmente (por exemplo: por que algumas partes do problema são resolvidas com RegEx?)
2. Método: como a solução funciona, de maneira tal que possa ser re-implementada por outro grupo.
3. Resultados: como o grupo avaliou que sua solução funciona.

O que **não** colocar no relatório:

1. Frases especulativas. Toda frase do relatório deve ser uma citação da literatura, ou uma observação direta, ou o resultado de um raciocínio, ou parte do senso comum.
2. Superlativos e floreios: “as gramáticas regulares são *essenciais* para...” ou “o experimento foi um *sucesso*!”
3. Demonstrações de teoremas e outros raciocínios muito longos: nesse

caso, é melhor citar a conclusão que está usando e indicar a referência de onde tirou.

4. Fatos que são corretos, mas que não têm nenhuma ligação com o trabalho em si: “a decisão sobre a validade de strings em gramáticas regulares pode ser resolvida usando autômatos finitos”.

Datas importantes

10/abril: dia de conferência sobre o trabalho. Neste dia, cada grupo deve mostrar à sala como funciona sua proposta de solução. Os grupos deverão fazer perguntas e pedir esclarecimentos uns aos outros, e o objetivo da sala toda é que todos saiam da sala, nesse dia, com um plano consistente para a solução do trabalho.

24/abril: faremos um dia de revisão do relatório em sala. Cada dupla deve trazer duas cópias físicas do seu relatório, pois analisaremos os relatórios uns dos outros.

03/maio: entrega final do código e do relatório. Este relatório deverá incorporar as modificações propostas na aula do dia 24/abril. O prazo de entrega é às 23h59min.

Notas

O trabalho receberá nota em três quesitos:

- 30% da nota será referente à participação no dia de discussão. A nota será dada da seguinte forma:
 - 10 – se o grupo apresentou uma proposta factível de solução ou encontrou uma solução factível durante a aula (a tempo de apresentar), e entende as ferramentas ou técnicas que deverão ser usadas para solucionar os problemas principais que serão encontrados. O grupo é responsável por identificar esses problemas.

- 5 – se o grupo apresentou um esboço de solução, mas que ignora problemas importantes, que podem trazer problemas ao grupo no futuro.
- 0 – se o grupo não apresentou proposta de solução ou se a proposta de solução apresentada está fora do tema.
- 35% da nota será referente ao relatório final, conforme a grade de correção divulgada e conhecida por todos. Haverá um dia de revisão do relatório.
- 35% da nota será referente ao código, que inicia em 10 e considera os seguintes descontos:
 - -10 se o programa não compila ou não executa (ele será testado em máquinas Linux!)
 - -4 para cada versão não implementada (devem ser entregues todas as operações matemáticas de soma, subtração e outra a definir, além da precedência de parêntesis).
 - -2 se não há script de teste
 - -2 se não há instruções para execução
 - -2 se o programa tem bugs que poderiam ser resolvidos facilmente (exemplo: não inicializa registradores)
 - +2 se o comando *make test* compila o programa, executa automaticamente todos os testes e imprime na tela os resultados de todos os testes.

A nota final será a média ponderada das notas de cada etapa se todas forem maiores que 3. Caso alguma nota seja menor que 3, a nota do projeto será igual à menor nota entre as etapas.