

Trabalho Prático - Especificação da Etapa 6: Geração de Código *Assembly*

Resumo:

Na sexta etapa do trabalho de implementação de um compilador você deve implementar a função que traduz o código intermediário gerado na etapa anterior para código *assembly*, ou seja, em linguagem com mnemônicos para ser traduzido para executável por um montador.

Tarefas necessárias:

- Selecionar a arquitetura-alvo e o montador a ser usado. Aconselha-se o uso do montador do gcc para gerar assembler AT&T rodando em Linux;
- Implementar função `generateAsm()` que escreve o programa em assembly em um arquivo de saída. Esta função deve imprimir as partes fixas do código no início, percorrer a Tabela de Símbolos para imprimir a declaração de cada uma das variáveis e constantes (literais) do programa, e finalmente, e mais importante, percorrer a lista de TACs gerada na etapa anterior e imprimir um pequeno conjunto de instruções ASM (tipicamente de 1 a 4) para implementar cada instrução do código intermediário;

Controle e organização do seu código fonte

Você deve seguir as mesmas regras das etapas anteriores para organizar o código, permitir compilação com **make**, permitir que o código seja rodado com **./etapa6**, e esteja disponível como **etapa6.tgz**. Nesta etapa o aluno deve incluir um pequeno relatório em arquivo `‘.txt’` ou `‘.pdf’` descrevendo as ferramentas e comandos empregados e mostrando *logs* de funcionamento de programas de teste, depois de compilados pelo seu compilador, rodando. Os alunos devem estar disponíveis para demonstração do trabalho rodando, individualmente ou em grupo, em horários a serem combinados com o professor.

Porto Alegre, Setembro de 2020