

**2º Trabalho Prático**  
CIC 116432 – Software Básico  
Prof. Diego de Freitas Aranha  
2º Semestre de 2013

## 1 Introdução

O trabalho consiste em implementar um método de tradução entre dois conjuntos de instruções distintos, produzindo como resultado um programa executável em uma arquitetura moderna.

## 2 Método

O método implementado deve traduzir código-objeto produzido a partir da linguagem de montagem simbólica apresentada em sala para um programa em *Assembly* em arquiteturas IA-32 compatível com a sintaxe do montador **nasm**. Opcionalmente, o método deve também produzir como resultado um programa executável no formato ELF.

O programa de tradução deve ser implementado na linguagem C e receber como parâmetro único em linha de comando o nome de um arquivo na forma `<arquivo>.bin`. Este arquivo deve conter o programa para a arquitetura hipotética como uma seqüência de palavras de 16 *bits*, respeitando o formato de saída do montador desenvolvido anteriormente. A partir desse arquivo, o programa de tradução deve *desmontar* o código-objeto e produzir dois programas:

- Um arquivo com nome `<arquivo>.s`, contendo a versão traduzida do programa para a linguagem de montagem IA-32 na sintaxe do montador **nasm**;
- Opcionalmente, um arquivo com nome `<arquivo>`, contendo o programa executável em formato ELF resultante da tradução.

Observe que as seções de texto e dados da linguagem de montagem hipotética devem ser convertidas para o novo formato de forma a conservar o comportamento correto do programa. Para eliminar ambiguidades, você pode considerar que a instrução **STOP** divide o programa em seções de texto e dados; e que o programa de entrada possui apenas uma instrução **STOP** localizada no final da seção de texto. Para tratar

o formato de arquivo ELF, é permitida a utilização de bibliotecas externas, como a `libelf`. É incentivada a utilização do montador `nasm` para montagem do arquivo `<arquivo>.s` resultante, com o objetivo de facilitar a detecção de erros na produção do programa executável. Idealmente, o arquivo executável `<arquivo>` deve ser bastante similar ou até idêntico ao programa produzido pelo montador `nasm` a partir do arquivo `<arquivo>.s`.

Por fim, as rotinas de entrada e saída podem utilizar os arquivos `io.mac` e `io.o` que acompanham a especificação como referência, mas precisam ser devidamente convertidas para seqüências equivalentes de instruções na arquitetura alvo, incluindo eventuais conversões de formato. Ou seja, o arquivo produzido pelo trabalho deve ser completamente auto-contido, sem dependências externas. Por simplicidade, você pode assumir que as entradas fornecidas serão sempre números inteiros bem-formados.

### 3 Avaliação

O prazo de entrega do trabalho é 20 de Dezembro de 2013. A entrega consistirá em:

- Código-fonte completo e comentado com instruções de compilação, incluindo possíveis bibliotecas externas;
- Programas de exemplo que demonstrem o funcionamento correto do programa.

A forma de entrega é por *e-mail*, de preferência utilizando o rótulo [SB]. Todas as entregas receberão mensagem de confirmação. O trabalho pode ser feito individualmente ou em dupla e será premiado com bonificação o trabalho que apresentar comportamento correto e produzir programas executáveis que apresentem melhor desempenho. A análise de desempenho será realizada pelo professor com programas de *benchmarking* suficientemente extensos. Desta forma, a bonificação por desempenho do trabalho é de no máximo 1 ponto, dividido em até 2 alunos diferentes, a serem aplicados exclusivamente na terceira prova. Um ponto adicional de bonificação será conferido aos trabalhos que implementarem corretamente a conversão para formato ELF nativo.