



Universidade de São Paulo - USP
Instituto de Ciências Matemáticas e de Computação – ICMC
Departamento de Sistemas de Computação – SSC
SSC0112 Organização de Computadores Digitais I
1o. Trabalho Prático – disponibilizado em 30/03/2017

Implementação de uma Tabela Hash em Assembly MIPS

O objetivo deste trabalho é implementar em Assembly MIPS uma Tabela Hash para armazenar números inteiros positivos. As opções previstas para esta Tabela Hash são: inserção, remoção, busca e visualização da tabela hash inteira, incluindo colisões. Estas quatro opções são acessadas via um menu, acrescido apenas da opção de finalização da aplicação, como última opção. A visualização da tabela deve ser por linhas, onde cada linha indica uma entrada da tabela e logo em seguida devem ser mostrados os valores conflitantes. A operação de busca deve retornar o valor da entrada hash que o elemento buscado se encontra ou -1 caso o elemento não foi encontrado.

A função hash a ser usada é o resto da divisão do número a ser inserido (chave) pelo tamanho da Tabela Hash. O tamanho da Tabela Hash é de **16 posições (tamanho fixo no código)**. O tratamento de colisões de cada posição da Tabela Hash deve ser feito com listas duplamente encadeadas (uma para cada posição). Os números inseridos nas listas que tratam as colisões das posições da Tabela Hash não precisam estar ordenados, podendo seguir a ordem de inclusão/remoção naquela respectiva posição. As listas são dinâmicas não têm um limite máximo para a quantidade de elementos.

Ao entrar nas opções de inserção, remoção e busca, o usuário deve fornecer uma chave e realizar a operação solicitada. Ao final destas operações, uma nova chave deve ser solicitada. O retorno ao menu de opções só ocorre quando o valor -1 (menos um) for fornecido. Deve haver verificação se a chave é um inteiro positivo. A opção de visualização retorna ao menu de opções após exibir a Tabela Hash atual completamente.

A correção levará em conta a execução correta do algoritmo e a qualidade do código fonte feito (qualidade == indentação, comentários corretos, nome significativos para rótulos, espaços entre porções do código, etc).

O trabalho deverá ser feito em grupo e submetido no Moodle até a data já definida em sala de aula (esta data está informada no Moodle).

Quaisquer características omissas a esta especificação deverão ser reportadas ao professor para orientação de como proceder na construção dos algoritmos.

Bônus:

Caso a lista para tratamento de colisões seja ordenada na inserção/remoção e a busca dela seja implementada como Busca Binária, o trabalho receberá um bônus proporcional à qualidade desta solução proposta (este tratado como Atividade Complementar na avaliação).

Outras iniciativas positivas que estendam esta especificação também poderão ser consideradas como bônus. A avaliação da pertinência ao bônus será feita pelo professor responsável pela disciplina.