

Universidade Federal da Bahia
Graduação em Ciência da Computação
MATA54 - Estruturas de Dados e Algoritmos II
Primeiro Trabalho Prático
Prof. Flávio Assis
Semestre 2017.1 - 12 de junho de 2017

Resolução de *Hashing* com Encadeamento

1 Descrição Geral do Trabalho

Neste trabalho o aluno implementará um arquivo estruturado como uma tabela *hash*. O método de resolução de colisão a ser implementado deverá usar encadeamento, como apresentado em sala de aula. Em particular, o método encadeia registros que contêm o mesmo endereço original (valor da função de *hashing*).

Os dados do usuário a serem armazenados no arquivo são: uma *chave*, de valor inteiro não negativo; uma cadeia de, no máximo, 20 caracteres, que irá armazenar um *nome*; e um outro valor inteiro não negativo, que irá armazenar uma *idade*. O programa deverá conter uma constante definida com o seguinte identificador:

TAMANHO_ARQUIVO: indica o número máximo de registros do arquivo.

O valor inicial da constante TAMANHO_ARQUIVO deve ser 11. O programa deve ser feito de forma que o valor desta constante possa ser modificado.

A função de *hashing* a ser utilizada, denominada h , é:

$$h(chave) = chave \bmod \text{TAMANHO_ARQUIVO}$$

Observações importantes: O programa deve manter as atualizações em arquivo. A correção levará em consideração que o estado dos dados é persistente. Com isto, um teste pode ser feito, por exemplo, inserindo-se um registro, terminando a execução do programa e fazendo uma consulta ao registro em nova invocação do programa. Neste caso o registro deve ainda estar no arquivo.

Adicionalmente, lembre-se de que é assumido que a memória principal é insuficiente para armazenar todos os dados. Portanto, por exemplo, uma implementação que mantenha a estrutura do arquivo em memória (em um vetor, por exemplo) e o salva por completo no arquivo será considerada inaceitável.

O arquivo deve ser armazenado em formato binário.

2 Formato de Entrada e Saída

A entrada conterá uma sequência de operações sobre o arquivo. As operações e seus formatos estão descritos abaixo:

1. **insere registro:** esta operação conterá quatro linhas. A primeira linha conterá a letra 'i'. A segunda conterá um valor de chave. A terceira conterá uma sequência de até 20 caracteres, que corresponderá ao campo *nome*. A quarta linha conterá um valor de idade. A sequência de caracteres da terceira linha conterá qualquer sequência de letras (minúsculas, sem acento, nem cedilha) e espaços, sendo que o primeiro e último caracteres não serão espaço.

Esta operação verifica se já há registro no arquivo com o valor de chave indicado. Se sim, esta operação gera na saída, em uma mesma linha, a sequência de caracteres '*chave ja existente:*', seguida de um espaço, seguido do valor da chave. Se a chave não existir, a operação insere o registro no arquivo, sem gerar saída.

2. **consulta registro:** esta operação conterá duas linhas. A primeira linha conterá a letra 'c'. A segunda conterá um valor de chave.

Se houver registro no arquivo com o valor de chave indicado, esta operação gera na saída a sequência de caracteres '*chave:*', seguida de um espaço, seguido do valor da chave. Em seguida, na próxima linha escreve o valor do nome associado ao registro, e, na linha seguinte, o valor da idade associada ao registro. Se não houver registro no arquivo com o valor de chave indicado, esta operação gera na saída a sequência de caracteres '*chave nao encontrada:*', seguida de um espaço, seguido do valor da chave.

3. **remove registro:** esta operação conterá duas linhas. A primeira linha conterá a letra 'r'. A segunda conterá um valor de chave.

Se houver registro no arquivo com o valor de chave indicado, esta operação causará a remoção do registro e não gerará saída. Se não houver registro no arquivo com o valor de chave indicado, esta operação gera na saída a sequência de caracteres '*chave nao encontrada:*', seguida de um espaço, seguido do valor da chave.

4. **imprime arquivo:** esta operação conterá apenas uma linha, contendo a letra 'p'. Esta operação imprimirá o arquivo, da forma a seguir. Os registros serão apresentados, um em cada linha, em ordem, do registro de índice 0 até o registro de índice *TAMANHO_ARQUIVO* - 1. Cada linha conterá: o índice do registro, seguido de dois pontos (':'), seguido de um espaço. Se a posição do arquivo estiver vazia, a sequência de caracteres '*vazio*' deverá ser apresentada, seguida de um espaço, seguido do valor do campo apontador. Se a posição estiver ocupada, deve ser apresentada a chave do registro, seguida de um espaço, seguida da sequência de caracteres (nome), seguida de um espaço, seguido da idade, seguida de um espaço, seguido do valor do campo apontador. Em ambos os casos (posição vazia ou não), se o campo apontador tiver valor nulo, deve ser impressa a sequência de caracteres '*nulo*'.

5. **média de acessos a registros do arquivo:** esta operação conterá apenas uma linha, contendo a letra 'm'. Esta operação apresenta, em uma

linha, apenas o valor da média do número de acessos a registros do arquivo, considerando-se uma consulta a cada um dos registros armazenados no arquivo. Esta média deve ser apresentada sempre como um valor real, com uma única casa decimal.

6. **término da sequência de comandos:** a sequência de comandos será terminada por uma linha com a letra 'e'.

Importante: o programa não deve gerar nenhum caractere a mais na saída, além dos indicados acima. Em particular, o programa não deve conter menus.

Não deve haver espaço entre linhas na saída. A saída deve apresentar os caracteres em letras minúsculas.

3 Observações

Trabalho individual.

Data de entrega: 07/07/2017

Linguagens de programação permitidas: C, C++, Java ou Python.

Observação Importante: Para as linguagens C, C++ e Java, somente trabalhos feitos utilizando-se os seguintes compiladores serão aceitos:

- C: gcc ou djgpp
- C++: g++ ou djgpp
- Java: compilador java recente, disponibilizado pela Oracle.

Não serão compilados trabalhos em outros compiladores! Erros ocasionados por uso de diferentes compiladores serão considerados erros do trabalho!

O aluno deverá armazenar submeter seu trabalho através do *moodle*.