

Atividade – Tabela Hash 2

Terminem a segunda parte da implementação iniciada na aula do dia 22/11/2017 sobre tabela Hash.

A implementação poderá ser feita em grupo de até 3 pessoas.

Enviem todos os códigos desenvolvidos para o STOA (Link: [Tarefa - Tabela-Hash 2](#)).

Não se esqueça de colocar nos códigos os nomes de todos os integrantes do grupo.

Se for compactar o código coloque a extensão **.zip** ou **.rar**.

Data de Entrega: **06/12/2017** até as **21:00** via STOA.

Criação de Índice Remissivo (Continuação)

Implemente em C um conjunto de funções para inserção de dados em uma Tabela Hash. Você deverá implementar a abordagem de lista encadeada para tratar colisões. Nessa implementação você deverá criar uma lista encadeada para cada uma das posições da tabela Hash. Ao inserir um elemento na hash você deverá inserir esse elemento no início da lista encadeada, isso irá garantir que a operação de inserção executará em $O(1)$.

Problema de criação de índice remissivo: Várias aplicações necessitam de um relatório de referências cruzadas. Por exemplo, a maioria dos livros apresentam um índice remissivo que corresponde a uma lista alfabética de palavras chave ou palavras relevantes do texto com a indicação dos locais no texto onde cada palavra chave ocorre.

Como exemplo, suponha um arquivo contendo um texto constituído como abaixo (**vocês deverão ler os dados de um arquivo não pode estar diretamente no código o texto abaixo, para que no futuro esse texto possa ser mudado de forma fácil**) (Obs: as linhas escritas em vermelho não estão de fato no texto é apenas um indicativo para você saber a linha):

Linha 1: Good programming is not learned from
Linha 2: generalities but by seeing how significant
Linha 3: programs can be made clean easy to
Linha 4: read easy to maintain and modify
Linha 5: human-engineered efficient and reliable
Linha 6: by the application of common sense and
Linha 7: by the use of good programming practices

Assumindo que o índice remissivo seja constituído das palavras chave (**Vocês deverão fazer a leitura dessas palavras chave de um arquivo, para que no futuro essas palavras chaves possam ser mudadas de forma fácil**):

```
and
by
be
easy
human-engineered
programming
programs
to
```

o programa para criação do índice deve produzir a seguinte saída:

```
and -> 6, 5, 4
be -> 3
by -> 7, 6, 2
easy -> 4, 3
human-engineered -> 5
programming -> 7, 1
programs -> 3
to -> 4, 3
```

Note que a lista de palavras chave está em ordem alfabética (**observação: não é necessário fazer um algoritmo de ordenação para isso, a ordenação ocorreu porque as palavras da lista de palavras chave já estavam em ordem alfabética**). Adjacente a cada palavra chave está uma lista de números de linhas, um para cada vez que a palavra ocorre no texto.

Projete um sistema para produzir um índice remissivo como projetado acima. Adapte a função hash a esse novo contexto. Nesse exemplo, temos apenas 8 palavras chaves, dessa forma, será criada uma tabela hash com 8 posições, se existissem mais palavras chave então seriam necessárias mais posições na tabela.