



Laboratório 4 - Tabelas de Dispersão

Semana de 11 a 15 de Maio de 2015

Duração: 2 horas

O objectivo deste trabalho é a implementação de uma tabela de dispersão (hash-table), usando o método de separação em listas. A tabela será usada para armazenar palavras. O programa, a completar na aula, lê as palavras contidas num ficheiro e coloca-as na tabela de dispersão juntamente com o número de ocorrências das mesmas. Após a construção da tabela, o seu conteúdo (palavras distintas e respectivo número de ocorrências) deverá ser escrito num ficheiro **<nome_ficheiro_entrada>.palavras**.

Os ficheiros **hash.c** e **pal.c** contém já algum código auxiliar, designadamente a definição das estrutura e funções indicadas abaixo. Analise o código que lhe é fornecido.

Estude o código do programa que se encontra nos ficheiros **lab06.c**, **tree.c**, **tree.h** e **queue.h**:

- O código implementa uma estrutura de dados em lista, onde cada elemento contém uma referência para uma cadeia de caracteres e um inteiro para guardar o número de ocorrências.
- A função **AlocaNo(...)** recebe uma cadeia de caracteres, aloca e inicializa o nó.
- A função **AcrescentaNo(no1, no2)** tem argumentos do tipo apontador para No. A função acrescenta o nó apontado por **no1** à lista, colocando-o a seguir ao nó apontado por **no2**.
- A função **AlocaTabela(...)** tem um argumento do tipo inteiro, que corresponde ao número de elementos (**m**) da tabela. Os elementos da tabela são de tipo apontador para No.
- A função de dispersão **Disperse(...)**, aceita um argumento de tipo cadeia de caracteres (**s**) e um argumento do tipo inteiro (**m**). A função devolve um inteiro entre 0 e $m - 1$. O valor é calculado através do seguinte pseudocódigo:

```
Disperse(s, m)
    h = 0
    p = 117
    n = comprimento(s)
    para i entre 0 e n-1
        executar h = mod(s[i] * p + h, m)
    devolver h
```

1. Escreva uma função **InserirNaTabela(...)**, que insere uma cadeia de caracteres (passada em argumento) na tabela, de acordo com o índice produzido pela função de dispersão. Se a palavra já se encontrar na tabela, não é novamente inserida, mas o número de ocorrências é incrementado. Caso a palavra não exista na tabela, mas a função de dispersão forneça um índice já ocupado na tabela (ocorrência de colisão de indexação), insira o novo nó (com a nova palavra) no fim da lista.
2. As funções de leitura do ficheiro são fornecidas também no ficheiro **pal.c**. Modifique e acrescente as funções necessárias para o funcionamento correcto do programa (leitura, contagem e armazenamento das palavras de um ficheiro numa tabela de dispersão; escrita em ficheiro das palavras distintas e respectivas ocorrências). O programa recebe 2 argumentos na linha de comando: o nome do ficheiro de entrada e o valor do parâmetro **m**.
3. Experimente o seu programa com os ficheiros de texto que se encontram na sua área. Verifique o comprimento das listas criadas e o número de comparações de cadeias de caracteres efectuadas durante a inserção das palavras na tabela, para vários valores do parâmetro **m**:

- $m = 1$
- $m = 11$
- $m = 13$
- $m = 120$
- $m = 117$
- $m = 121$
- $m = 128$

Avalie e justifique os resultados.

4. Nos casos anteriores, e para os valores de m indicados anteriormente, determine experimentalmente o valor médio e a variância da distribuição de palavras pela tabela.