

Visão Geral

O fim principal do programa é conseguir buscar uma palavra em um arquivo, ambos fornecidos pelo usuário, e em seguida, caso a palavra tenha sido encontrada, armazenar em qual linha cada ocorrência da mesma se deu em uma tabela hash.

Obviamente o problema inicial era encontrar tal palavra no texto, ao mesmo tempo em que se obtia a informação de sua linha, todas as vezes que se repetisse. O segundo desafio era construir uma tabela hash para armazená-la junto a suas informações.

Seguindo o conceito de TAD foram criadas duas bibliotecas: text.h e hash.h.

text.h

Responsável por realizar a busca da palavra no texto, coletando a linha e a quantidade de ocorrências da mesma. Esta operação pode ser realizada através da função.

`Word getInfo(char fileName[], char word[]);` Que recebe o nome do arquivo e a palavra a ser buscada, retornando uma struct que contém as informações obtidas.

Ela também possui as funções:

`int getNWord(Word W);` Extrai a quantidade de ocorrências da palavra no texto.

`int *getWLines(Word W);` Extrai a linha de cada ocorrência da palavra no texto.

`void printWord(Word W);` Imprime na tela as informações da palavra em questão.

hash.h

Responsável por criar a tabela hash e manipulá-la.

Possui as seguintes funções:

`HashTab createHashTab(int Size);` Recebe o tamanho desejado para a tabela, e retorna um struct que a contém.

`Cell insertInTab(char word[], int wordLines[], int occurrences, HashTab H);` Recebe a palavra a ser armazenada, em quais linhas foi encontrada e o número de ocorrências. Retorna a célula da tabela em que ela foi armazenada.

`int hash(char word[], HashTab H);` Recebe uma palavra e a hash em que foi ou será armazenada, e calcula seu valor hash de acordo com a função especificada para este trabalho prático.

`Cell searchInTab(char word[], HashTab H);` Recebe uma palavra e uma hash. Retorna a célula em que esta palavra está armazenada nesta hash em casos de sucesso, e NULL caso contrário.

`void printCell(Cell C);` Recebe uma célula de alguma hash e a imprime na tela. Vale ressaltar que Chain Position é a posição da palavra na lista encadeada, e para cada ocorrência da palavra ela será impressa na tela novamente alterando apenas o valor de Line.

`void printHash(HashTab H);` Recebe uma hash e a imprime todas suas células na tela.

A realização do trabalho foi interessante no quesito de poder criar uma tabela Hash na prática em conjunto com manipulação de arquivos, além de trabalhar o conceito de TAD provando sua utilidade. A principal dificuldade foi encontrada na implementação da text.h ao buscar as palavras e ter de armazenar em que linhas se encontravam ao mesmo tempo.

Bibliografia:

Slide disponibilizados na disciplina AEDS 1

Backes, André (2012). Linguagem C Completa e Descomplicada.

Ime usp, Tabelas de dispersão (hash tables): <https://www.ime.usp.br/~pf/mac0122-2002/aulas/hashtable.html>"