UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO
INF01203 ESTRUTURAS DE DADOS (2015/1)
Adriano Carniel Benin (173464)

Trabalho prático

Porto Alegre, Junho de 2015

1 Introdução

O trabalho consiste em implementar um programa que faça indexação e busca de palavras em arquivo texto.

2 Funcionamento do programa

O programa inicia indexando o arquivo de entrada numa árvore AVL. Para isso o arquivo é lido linha por linha, quebrando em palavras usando como delimitadores qualquer caractere que não seja uma letra. Para cada palavra se insere um nodo contendo a palavra em si e uma lista com as linhas em que foi encontrada, caso a palavra já exista na árvore apenas é inserido a nova linha na lista.

Na fase de busca são lidas as palavras contidas no arquivo de consulta e cada uma é procurada na árvore usando uma busca padrão de árvores ABP. Quando a palavra é encontrada escrevemos no arquivo de saída a palavra e sua lista de linhas correspondente. Toda a fase de busca é medida usando a função clock() contida em time.h.

3 Estruturas Utilizadas

O objetivo do trabalho é otimizar o tempo de pesquisa das palavras. Para isso foi usada a seguinte estratégia:

1. Percorrer o texto de entrada armazenando cada palavra numa árvore AVL, usando a ordem lexicográfica.
2. No mesmo nodo foi armazenado uma lista simplesmente encadeada contendo as linhas em que foi encontrada a palavra.
3. Caso a palavra já exista na árvore apenas adicionamos a linha em que foi encontrada no nodo correspondente.

As estruturas do programa são apresentadas abaixo:

```
typedef struct NODE_T {
  void * data;
  int height;
  struct NODE_T *right;
  struct NODE_T *left;

} TREE_NODE;

typedef struct _list_node {
  void *data;
  struct _list_node *next;
} LIST_NODE;

typedef struct token_t {
  wchar_t * word;
  LIST_NODE * list;
} W_TOKEN;
```

TREE_NODE recebe um tipo de dado genérico, no caso um ponteiro para W_TOKEN. W_TOKEN contém a palavra em si e um ponteiro para uma lista encadeada LIST_NODE com as linhas em que ela se encontra.

4 Escolha das estruturas

O objetivo do trabalho era otimizar o tempo de busca. Como a busca é por palavras conseguimos definir uma relação de ordem e logo podemos realizar a busca eficientemente com uma ABP. Foi escolhida a estrutura de árvore AVL por representar o melhor balanceamento possível e portanto a maior velocidade na busca. Cada nodo recebe como dado um struct token_t, que contém uma string e uma lista de linhas.

Para armazenar as linhas foi usado uma lista simplesmente encadeada por permitir uma quantidade arbitrária de elementos. A escolha dessa estrutura não influencia na velocidade da busca pois é utilizada apenas para exibir as linhas após a palavra ter sido encontrada.