Universidade Federal de Goiás
Instituto de Informática
Curso de Bacharelado em Sistemas de Informação
Introdução à Programação 2020-1

Trabalho Final de Curso
Implementação de um Índice Invertido

Prof. Thierson Couto Rosa

1 Objetivo

Este trabalho visa aplicar funções relacionadas a criação, abertura, e manipulação de arquivos da linguagem C e também alocação e re-alocação de memória. O programa deve ser desenvolvido individualmente.

O trabalho consiste em escrever um um programa para:

a) ler um arquivo tipo texto contendo as descrições de documentos textuais de uma coleção;

b) criar um *índice invertido* a ser definido a seguir;

c) receber do teclado consultas formada por termos e mostrar listas de documentos onde todos os termos da consulta ocorrem.

Índice Invertido

Um índice invertido é uma estrutura de dados que permite encontrar rapidamente em uma coleção de texto, quais unidades da coleção um dado termo ocorre. Um exemplo de índice invertido é o *índice remissivo* que aparece (geralmente ao final) em livros didáticos e manuais técnicos. Nesse caso, podemos entender o livro como sendo uma coleção de páginas. O índice remissivo contém as principais palavras ou expressões do livro em ordem alfabética. Se uma pessoa quer saber em quais páginas (documentos) do livro (coleção), uma palavra ocorre, ela pode consultar a palavra no índice remissivo. Se a palavra estiver no índice, haverá ao seu lado, a sequência de páginas onde a palavra ocorre no livro. Veja o exemplo na figura1[1]

O índice invertido é uma estrutura de dados utilizada em diversas aplicações, tais como máquinas de busca (como Google, Bing), para classificação de documentos e para agrupamento de documentos. Essa estrutura permite encontrar rapidamente os documentos onde um termo ocorre, sem ter que percorrer exaustivamente todos os documentos, procurando pelo termo.

[1]Fonte: http://latexbr.blogspot.com/2013/01/indice-remissivo-no-latex.html.

Figura 1: Trecho de um índice remissivo

Consultas no Índice invertido

Uma vez construído um índice invertido para os documentos de uma coleção, consultas formadas por termos podem ser feitas rapidamente sobre o índice. A consulta mais simples é formada por apenas um termo. Nesse caso, o sistema de computação pesquisa primeiro pelo termo no índice e se o termo for encontrado, retorna a lista de documentos em que o termo ocorre. Porém, consultas podem ser mais complicadas, sendo formadas por mais de um termo e em alguns casos por operadores, formando expressões de consulta. Por exemplo, a expressão: $casa \vee mesa$, requer que todos documentos contendo "casa" **ou** ($\vee$) "mesa" sejam encontrados. A consulta $casa \wedge mesa$, requer que todos os documentos que contenham necessariamente os dois termos sejam apresentados.

A construção de índices invertidos e o processamento de consultas, bem como problemas de *ranking* de documentos para atender a consultas são problemas estudados em uma área da Computação denominada *Recuperação da Informação* (*Information Retrieval*).

2 Descrição do Trabalho

2.1 Descrição do arquivo de Entrada

A coleção de documentos já foi processada e foi transformada em um único arquivo tipo texto, onde cada linha do arquivo corresponde a um documento da coleção. Cada linha tem o seguinte formato:

$$c\ t_1{:}ft_1\ t_2{:}ft_2\ t_3{:}ft_3\ \cdots\ t_n{:}ft_n$$

O primeiro valor c de uma linha corresponde a um número inteiro indicando a categoria a qual o documento pertence (esportes, cultura, Matemática, etc). Após o valor de classe há uma sequência de pares do tipo $t_i{:}ft_i$, onde t_i é um número inteiro correspondente a uma palavra (ou termo) que aparece no documento e ft_i é a frequência com que o termo t_i aparece nesse documento (linha). Por exemplo, suponha que na j-ésima linha do arquivo (correspondendo ao j-ésimo documento), aparece o seguinte texto:

5 0:13 1:1 6:1 10:1 13:2 15:2 17:1 18:2 26:16 27:3 28:17 29:2 30:3 36:3 37:54 40:21

Essa linha indica o seguinte: o documento j é da categoria 5 e é composto pelos termos indicados pelos primeiros componentes dos pares seguintes. Por exemplo, o termo 0 ocorre 13 vezes nesse documento, o termo 1 ocorre apenas uma vez e os termos 3, 4 e 5 não ocorrem no documento j. Repare que os pares aparecem ordenados em ordem crescente dada pelo valor dos termos.

2.2 Módulos do Trabalho

O trabalho pode ser subdividido funcionalmente em três módulos ou partes: a) o módulo de construção do índice invertido; b) o módulo de processamento de consultas e c) o módulo de liberação do índice invertido O módulo de construção do índice invertido deve ser executado primeiro antes que consultas possam ser lidas e processadas.

Módulo de construção do Índice Invertido

O módulo de construção do índice invertido deve receber como entrada o arquivo contendo a coleção de documentos e gerar o índice invertido. A figura 2 ilustra como seria o índice invertido criado.

O vetor de cor azul corresponde ao *vetor de termos*. Cada elemento do componente vetor de termos aponta para uma sequência de triplas, onde cada tripla contém os seguintes componentes: a) d_i que indica o número de um documento (linha do arquivo de entrada) onde o termo ocorreu, b) f_{t,d_i} que é a frequência com que o termo ocorreu no documento d_i e c) a classe c do documento d_i. Uma sequência de triplas é denominada *lista invertida*. Cada vetor horizontal na figura 2 corresponde a uma lista invertida.

A seguir são discutidas algumas possibilidades para construção do índice invertido. Ambas usam alguns componentes básicos em comum: a) uma struct para representar cada tripla, que denominaremos aqui de *tipoTripla* apenas para facilitar a discussão ; b) o vetor de termos que inicialmente pode ser definido como um vetor de ponteiros para *tipoTripla*. Pode-se usar o número inteiro que representa um termo no arquivo como índice para acesso direto ao vetor de termos. Assim, se encontramos no arquivo texto um par 0:2, podemos usar o termo 0 para acessar o índice 0 do vetor de termos.

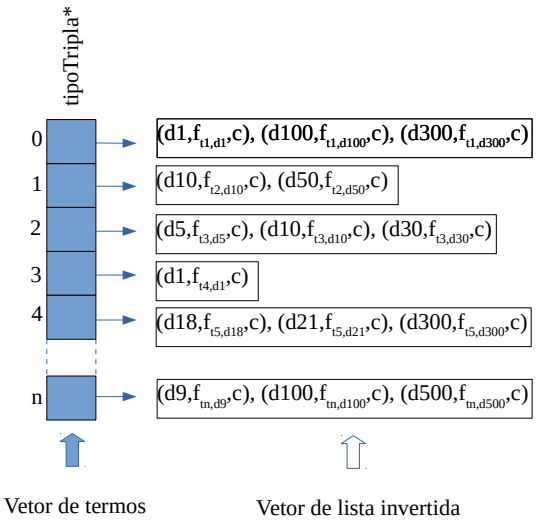

Figura 2: Ilustração correspondente ao índice invertido para a coleção de documentos.

No momento da leitura do arquivo com a coleção, ainda não se sabe quantos termos existem na coleção. Portanto, nesse momento da construção do índice não há como definir o tamanho do vetor de termos (vetor azul na figura). Logo, há duas formas de criar esse vetor. A primeira, consiste em fazer uma leitura no arquivo da coleção apenas com o objetivo de encontrar o maior valor de termo. Suponha que o maior valor de termo no arquivo corresponda a t_{max}. Após essa leitura, aloca-se dinamicamente, de uma única vez, o vetor de termos como um vetor de ponteiros para tipoTripla com tamanho igual a $t_{max} + 1$. Em seguida lê-se novamente o arquivo, criando listas invertidas para cada termo, à medida que as triplas são formadas durante a segunda leitura do arquivo.

Outra possibilidade é criar o índice invertido fazendo uma única leitura no arquivo da coleção. Nesse caso, pode ser criado um vetor de termos dinamicamente e gradativamente. Inicialmente cria-se dinamicamente um vetor de termos de tamanho x, onde x é um valor inicial definido pelo aluno (ex. 20.000). Enquanto não for encontrado um termo de valor maior que $x - 1$, pode-se usar o vetor alocado. Entretanto, quando for encontrado um termo de valor maior que $x - 1$ é necessário realocar o vetor de termos (usando *realloc()*). A realocação pode ser por uma área de tamanho igual ao valor do termo encontrado mais 1 ou pode ser por um valor maior do que essa soma.

Embora existam as duas formas de criar o vetor de termos, como descrito acima, o mesmo

não ocorre para as listas invertidas. Os vetores que formam as listas invertidas necessitam ser alocados e re-alocados dinamicamente, pois não há como saber de antemão em quantos documentos cada termo ocorre. Quando uma nova tripla tiver que ser inserida em um vetor de lista invertida, é preciso primeiro verificar se a inserção não extrapola o tamanho do vetor. Para isso, é necessário que se armazene para cada vetor de lista invertida, a informação sobre seu tamanho atual. Como o tamanho da lista invertida é uma informação associada ao termo correspondente à lista invertida, uma boa sugestão é armazenar essa informação no próprio vetor de termos. Assim, muda-se o tipo do vetor de termos. Em vez de ser um vetor de ponteiros para *tipoTripla*, define-se um novo tipo (por ex. tipo vetTermos) que é uma struct e o vetor de termos passa a ser definido como um vetor desse tipo. Essa struct possui dois campos: um campo do tipo *int* que armazena a informação sobre o tamanho da lista invertida correspondente e um campo ponteiro para tipoTripla que armazena o endereço de início do vetor lista invertida. A figura 3 ilustra essa nova configuração para o índice invertido.

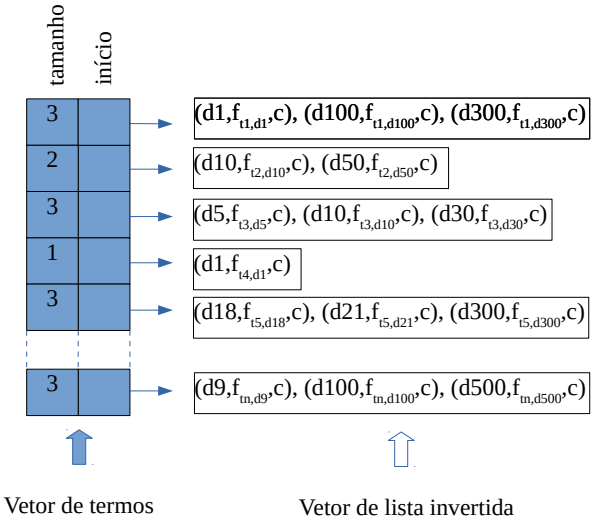

Figura 3: Vetor de termos composto por dois campos: tamanho e ponteiro para o vetor dinâmico que forma a lista invertida.

Módulo de consulta

Uma vez que o índice invertido tiver sido criado, o programa deve chamar o módulo de consulta. Nesse módulo, deve ser apresentada uma tela ao usuário com uma pergunta do tipo "Se quiser fazer uma consulta digite 1, senão, digite 0". Se o usuário digitar 1, a tela deve ser limpada e uma nova tela com a seguinte frase deve aparecer: "Digite os números de termos correspondentes a uma consulta (no máximo 5 termos), separados entre si por um espaço". Em seguida, o módulo deve ler uma linha com os números correspondentes aos termos da consulta. Deve acessar o índice invertido para cada termo da consulta e mostrar a interseção de todas as listas entre si. Caso a interseção for vazia, a seguinte mensagem deve ser emitida: "Não há documentos que contenham concomitantemente todos os termos pesquisados".

Após responder a uma consulta, o módulo de consulta deve apagar a tela e mostra a tela inicial novamente. Essa repetição de telas deve ocorrer até que o usuário digite 0 indicando

que ele não quer fazer mais consultas. Nesse caso o programa deve chamar o módulo de liberação do índice e terminar.

Módulo de liberação do índice invertido

Esse módulo deve ser chamado antes de terminar o programa e seu objetivo é o de liberar a área ocupada pelos componentes do índice invertido (vetor de termos e vetores de listas invertidas). Esse módulo deve primeiramente percorrer o vetor de termos e para cada elemento do vetor de termos, liberar a lista invertida apontada pelo elemento. Após esse percurso, o próprio vetor de termos deve ser liberado.

3 Avaliação

Se o funcionamento do programa descrito acima for implementado, o(a) aluno(a) terá nota máxima no trabalho. Porém, pontos extras na nota podem ser alcançados se o(a) aluno(a) implementar outros recursos extras no programa. Algumas sugestões possíveis para o melhoramento do programa incluem:

- aumentar os tipos de consultas que possam ser feitas. Na especificação acima, é feita uma interseção das listas invertidas correspondentes aos termos da consulta. Essa interseção implementa a operação **E** também denotada pelo símbolo $\wedge$ entre os termos da consulta. O trabalho pode ser expandido para implementar a operação **OU** ($\vee$) que corresponde a retornar a união das listas invertidas dos termos da consulta.

- Outra melhoria no programa é o armazenamento em disco do índice invertido e sua posterior leitura do disco. Assim, evita-se que o programa crie o índice invertido (uma operação mais lenta) toda vez que for executado. Se o índice invertido foi criado em uma execução anterior, basta lê-lo do disco e carrega-lo em memória.

O trabalho deve ser implementado individualmente e deve ser apresentado no horário das aulas em um dos dias: 12/01/2021 ou 14/01/2021 pelo(a) aluno(a). Os pontos extras conseguidos no trabalho serão utilizados para aumentar a nota de uma das provas do(s) aluno(s) que tenha recebido nota inferior a dez.

OBSERVAÇÃO IMPORTANTE

Será dada nota zero para os trabalhos que apresentarem cópias de trechos de código de trabalhos de colegas do semestre ou de outros trabalhos desenvolvidos em semestres anteriores.

4 Material de apoio ao desenvolvimento

No site da disciplina, na plataforma Turing há um link para um conjunto vídeos e slides que explicam o uso de arquivos. Nesse link se encontra também o arquivo contendo a coleção de documentos para a qual o índice invertido deve ser construído.