ACH2003 – Computação Orientada a Objetos - Primeiro semestre de 2013
Professores: Karina Valdivia Delgado e Flávio Luiz Coutinho

Aplicativo de busca de palavras em textos

Descrição geral do trabalho

Este trabalho consiste no desenvolvimento de um aplicativo para a realização de buscas de palavras em textos. Os textos sobre os quais as buscas serão realizadas devem estar contidos em arquivos .txt (*plain text*) e estes arquivos devem ser especificados em um arquivo de configuração carregado pelo aplicativo ao iniciar. O objetivo deste trabalho é empregar os conhecimentos adquiridos ao longo disciplina para que o aplicativo, bem como os mecanismos de busca implementados, sejam eficientes e bem projetados.

Durante seu funcionamento o aplicativo deve permitir, através de sua interface, que o usuário entre com uma palavra a ser pesquisada nos textos, e deve devolver como saída as seguintes informações:

1. Se a palavra em questão foi encontrada, indicando a quantidade total de ocorrências da mesma.
2. Uma listagem dos arquivos nos quais a palavra foi encontrada, com a quantidade total de ocorrências por arquivo.
3. Opção de visualização de cada arquivo da listagem anterior. Nesta visualização cada ocorrência da palavra procurada pelo usuário deve aparecer destacada no texto e deve haver um mecanismo de navegação para que o usuário possa avançar ou retroceder pelas palavras destacadas.
4. O tempo total que a busca levou para ser feita.

Mecanismos de busca

Uma etapa importante do desenvolvimento deste trabalho refere-se à implementação dos mecanismos de busca, responsáveis por encontrar as ocorrências da palavra pesquisada no conteúdo dos arquivos de texto gerenciados pelo aplicativo. O mecanismo de busca mais trivial é aquele que faz uma busca sequencial pelo conteúdo dos arquivos, a procura da palavra desejada. Apesar de simples, esta forma de busca não é muito eficiente, principalmente quando o usuário deseja realizar várias buscas durante uma mesma sessão de uso do aplicativo.

Uma maneira de se implementar buscas de forma eficiente é através do uso de índices. Assim como índices de livros nos permitem localizar capítulos e seções de forma rápida e sem que haja a necessidade de se folhear o texto de forma sequencial, o mesmo conceito pode ser aplicado na implementação de um mecanismo de busca. Neste caso, deve-se manter algum tipo de estrutura de dados que permita, de forma eficiente, saber se a palavra procurada existe nos textos gerenciados pelo aplicativo, em quais arquivos a palavra ocorre e a localização de cada ocorrência dentro de cada arquivo.

Neste trabalho você deve implementar pelo menos **dois mecanismos de busca**: a busca sequencial e pelo menos um mecanismo de busca mais eficiente que empregue algum tipo de índice. A ideia é que se possa fazer comparações de performance entre os diferentes mecanismos de busca implementados (por isso que o tempo total que a busca leva para ser feita faz parte da saída fornecida pelo aplicativo). Observe que a medição do tempo de uma busca deve considerar apenas o tempo da busca em si, e não o tempo gasto em eventuais operações subsequentes como exibição de resultados, atualização da interface, etc.

A fim de que as comparações de performance sejam justas, todos os dados necessários para que a busca seja feita devem ser previamente carregados em memória. No caso da busca sequencial isso significa que o conteúdo de todos os arquivos deve estar carregado para que uma busca possa ser feita. Já no caso de um mecanismo de busca baseado em índice, é o índice que deve estar carregado em memória em sua totalidade.

No caso da implementação do mecanismo de busca baseado em índice, o índice deve ser persistido em disco. Desta forma, o índice pode ser utilizado em outras sessões de uso do aplicativo, sem a necessidade de gerá-lo novamente (a não ser que o índice fique desatualizado). É papel do aplicativo fazer o gerenciamento do índice. Esse gerenciamento deve se dar da seguinte forma:

- Caso já exista e esteja atualizado, o índice deve ser carregado sempre que o aplicativo iniciar para que possa ser usado nas pesquisas feitas pelo usuário.
- Caso não exista, ou exista e esteja desatualizado, o índice deve ser (re)criado no inicio da execução do aplicativo. Sempre que o índice for (re)criado ele deve ser persistido em disco.
- O índice é considerado desatualizado quando:
  - Houver alguma modificação na lista de arquivos gerenciados (adição de novos arquivos ou remoção de arquivos).
  - Houver modificações em algum dos arquivos gerenciados pelo aplicativo.

A escolha do mecanismo de busca a ser utilizado durante a execução do aplicativo também deverá ser definida no arquivo de configuração.

Fases do trabalho

Este trabalho é dividido em duas fases. A primeira fase consiste na implementação dos mecanismos para busca de palavras em textos. Na segunda fase o foco será na implementação de uma interface gráfica e revisão do que foi produzido na primeira fase com a aplicação de padrões de projetos. Apesar de a primeira fase não exigir uma interface gráfica, ela deve oferecer uma interface minimal em modo texto.

A interface em modo texto deve oferecer as mesmas saídas que as que serão implementadas na versão final do aplicativo (especificadas na seção “Descrição geral do aplicativo”), com exceção da saída especificada no item (3). Para a primeira fase, a saída descrita no item (3) pode ser simplificada da seguinte forma: quando o usuário optar por “visualizar” um dos arquivos que satisfaz a busca, a saída deverá indicar o número das linhas nas quais existe a palavra procurada, bem como exibir as linhas correspondentes do arquivo.

De forma mais detalhada são esperados os seguintes itens em cada fase:

- Primeira fase:
  - Implementação da busca sequencial.
  - Implementação de um mecanismo de busca mais eficiente, que faça uso de índice.
  - Implementação da interface mínima em modo texto.
  - Relatório de até **cinco páginas** descrevendo a implementação dos mecanismos de busca e a modelagem de classes usadas no aplicativo. Deve-se explicar e justificar a escolha do mecanismo de busca baseado em índices que foi escolhido, apresentando uma análise de seu desempenho e compará-lo com a busca sequencial.

- Segunda fase:
  - Implementação da interface gráfica.
  - Revisão do que foi desenvolvido na primeira fase com a aplicação de pelo menos **três padrões de projetos**. Não conta os padrões já usados pelo próprio Java.
  - Relatório de até **dez páginas** sobre a implementação da interface gráfica e aplicação dos padrões de projetos no desenvolvimento do aplicativo (incluir o diagrama de classes completo do sistema e um para cada padrão usado). Deve-se explicar e justificar o uso dos padrões de projeto utilizados na segunda fase do trabalho. O relatório da segunda fase deve conter ainda uma avaliação experimental dos dois mecanismos de busca oferecidos pelo aplicativo em termos de tempo gasto para fazer a busca em si
    Além disso, calcular o tempo gasto para criar o índice e o tempo gasto em serializar e restaurar o índice.
    Na avaliação pode gerar arquivos aleatórios usando http://br.lipsum.com. Por exemplo pode criar 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 arquivos aleatórios de 1000000 de palavras cada um.

Parte opcional para a segunda fase (valendo bônus na nota)

Uma parte opcional da segunda fase do trabalho consiste em implementar a geração do índice de forma paralela. Considerando que vários arquivos de texto podem ser especificados no arquivo de configuração, uma potencial forma de acelerar o processo de geração do índice seria pela paralelização do processamento dos arquivos de texto. Por exemplo, pode-se gerar um índice intermediário para cada arquivo de forma paralela, e em seguida se fazer a junção destes índices intermediários para se chegar ao índice final (e este processo de junção dos índices intermediários também poderia ser paralelizado). Vocês tem total liberdade para decidir como implementar este processamento paralelo. Não esqueçam de documentar a solução escolhida no relatório caso optem por implementar esta parte opcional.

Outras informações

Este trabalho pode ser feito por grupos de até 4 alunos.

Se durante a segunda fase do projeto, por qualquer motivo, o grupo quiser implementar um mecanismo de busca baseado em índice diferente daquele usado na primeira fase, não há qualquer problema. Neste caso, contudo, o grupo deve apresentar no segundo relatório a análise teórica de desempenho para o novo mecanismo escolhido (além, é claro, do que já é pedido para o relatório da segunda fase).

Datas de entrega

- Primeira fase: 11 de Maio
- Segunda fase: 18 de Junho